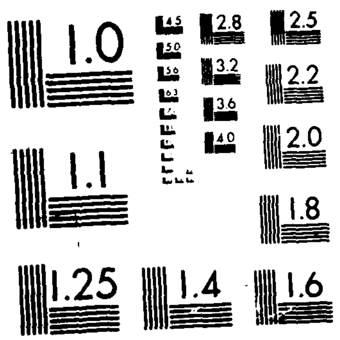
AD-A194 623 AUTOMATIC CLASSIFICATION OF DIGITALLY MODULATED SIGNALS 1/2
(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
SCHOOL OF ENGINEERING M P DESIMIO DEC 87
UNCLASSIFIED AFIT/GE/ENG/87D-15 F/G 12/9 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

AUTOMATIC CLASSIFICATION

OF

DIGITALLY MODULATED SIGNALS

Martin P. DeSimio
Civilian, Department of Defense

AFIT/GE/ENG/87D-15

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

UNCLASSIFIED

A194623

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE –	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/87D-15	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering 6b. OFFICE SYMBOL (If applicable): AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
See Box 19

12. PERSONAL AUTHOR(S)
Martin P. De Simio, B.S., Civilian, Department of Defense

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM ______ TO ______	1987 December	124

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Digital Modulation, LMS Algorithm, classification, Pattern recognition, binary phase shift keying
25	02		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

TITLE: AUTOMATIC CLASSIFICATION OF DIGITALLY MODULATED SIGNALS

Thesis Chairman: Glenn E. Prescott, Major, USAF
Instructor of Electrical Engineering

Approved for public release: IAW AFR 190-1.
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Glenn E. Prescott, Major, USAF	22b. TELEPHONE (Include Area Code): (513) 255-3576 22c. OFFICE SYMBOL: AFIT/ENG

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

This experiment investigates the performance of an adaptive technique for the classification of the following types of digitally modulated signals: binary amplitude shift keying (BASK), binary phase shift keying (BPSK), quaternary phase shift keying (QPSK), and binary frequency shift keying (BFSK).

The feature extraction process uses the mean and variance of the signal, and magnitudes and locations of the maxima in the spectrum of the signal, the spectrum of the signal squared, and the spectrum of the signal raised to the fourth power. The process of raising the signal to the second and fourth power and searching for narrowband energy near twice and four times the intermediate frequency is shown to provide useful information for the classification of BPSK and QPSK signals.

A computer simulation is performed to measure the properties of the classifier. First, the classifier is trained with a set of feature vectors calculated from 20 dB SNR signals. The Least Mean Squares (LMS) algorithm is the adaptive procedure used to generate the weight vectors used to form the linear decision functions. After training, these weight vectors are used to classify unknown signals from the signal set. One signal from each class at 20, 15, 10, and 5 dB SNRs are presented to the classifier. This method correctly classified all the signals considered during this experiment. However, the conclusiveness of the results are limited due to the small number of trials performed. The most important result is the discovery of features useful for the identification of M-ary PSK signals. Further study is recommended in this area.

AFIT/GE/ENG/87D-15

AUTOMATIC CLASSIFICATION
OF
DIGITALLY MODULATED SIGNALS

Martin P. DeSimio
Civilian, Department of Defense

AFIT/GE/ENG/87D-15

AFIT/GE/ENG/87D-15

AUTOMATIC CLASSIFICATION OF DIGITALLY MODULATED SIGNALS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Martin P. DeSimio, B.S.

Civilian, Department of Defense

December1987

Approved for public release; distribution unlimited

Preface

This thesis presents a method for the automatic identification of certain classes of digitally modulated signals by use of linear decision functions generated by the LMS algorithm. Although the set of unknown signals tested is limited, a new feature for the identification of phase shift keyed signals has been found.

Certainly, this thesis is the result of the efforts of many people. First of all, I would like to acknowledge the support of my thesis advisor, Major Glenn E. Prescott. Many interesting discussions were held during the course of this project; I am particularly grateful that he introduced me to the field of adaptive signal processing. The second person who requires acknowledgement is Mr. Vic Hanus of the Foreign Technology Division. He was the source of a number of useful suggestions concerning the application of the LMS algorithm. I would like to specially thank Mr. Rich Abrams of Antioch University for his assistance in producing the final draft of this thesis.

Also, I need to thank my family. Without the help of my parents and mother-in-law and father-in-law, this thesis would not have been written. They all provided support in ways too numerous to mention here. Finally, I must thank and congratulate my wife, Terese. She managed to care for me and our son, Patrick, and get us through the construction of a house while most of my time was devoted to school work.

Martin P. DeSimio

Table of Contents

List of Figures

List of Tables

Abstract

This experiment investigates the performance of an adaptive technique for the classification of the following types of digitally modulated signals: binary amplitude shift keying (BASK), binary phase shift keying (BPSK), quaternary phase shift keying (QPSK), and binary frequency shift keying (BFSK).

The feature extraction process uses the mean and variance of the signal, and magnitudes and locations of the maxima in the spectrum of the signal, the spectrum of the signal squared, and the spectrum of the signal raised to the fourth power. The process of raising the signal to the second and fourth power and searching for narrowband energy near twice and four times the intermediate frequency is shown to provide useful information for the classification of BPSK and QPSK signals.

A computer simulation is performed to measure the properties of the classifier. First, the classifier is trained with a set of feature vectors calculated from 20 dB SNR signals. The Least Mean Squares (LMS) algorithm is the adaptive procedure used to generate the weight vectors used to form the linear decision functions. After training, these weight vectors are used to classify unknown signals from the signal set. One signal from each class at 20, 15, 10, and 5 dB SNRs are presented to the classifier. This method correctly identifies all the signals considered during this experiment. However, the conclusiveness of the results are limited due to the small number of trials performed. The most important result is the discovery of features useful for the identification of M-ary PSK signals. Further study is recommended in this area.

AUTOMATIC CLASSIFICATION OF DIGITALLY MODULATED SIGNALS

I. Introduction

Background

Consistent identification of the modulation type of an unknown signals is not possible by human operators (Liedtke,1984:311). Applications such as radio spectrum surveillance and electronic warfare require automatic identification of the modulation type of the received signal (Chan and others,1985:22.5.1; Jondral,1985:177). The first application requires information on modulation type in order to demodulate the signal. The second application uses the information on modulation type in order to choose the appropriate electronic warfare strategy.

Problem and Scope

The purpose of the automatic signal classification method is to determine the modulation type of unknown signals. The set of signals to be considered for identification of modulation type are limited to forms of digital modulation. Specifically, the signals are binary amplitude shift keying (BASK), binary phase shift keying (BPSK), quaternary phase shift keying (QPSK), and binary frequency shift keying (BFSK).

The performance of the automatic classification procedure will be investigated by simulations with computer generated signals and noise. This procedure does not attempt to demodulate the unknown signals and is limited to a proof of concept of the classification method.

Summary of Current Knowledge

A review of unclassified literature from 1982 to 1987 reveals three references concerning the identification of the modulation type of signals. Two of the papers present similar approaches to the identification problem. The earlier of the two papers, which was written by Liedtke, provides the framework for a later paper by Jondral which is an extension of Liedtke's work. Liedtke's paper does not present a theoretical development of the statistics involved with the decision functions. However, Jondral uses an adaptive procedure which is trained by a learning process and is shown to be a form of classifier which minimizes the mean squared error. A third paper by Chan and others presents an approach for the identification of the modulation type of signals based upon the statistical properties of their envelopes. The three papers are summarized below.

Summary of Liedtke's Paper. The paper by Liedtke describes a method for the automatic classification of digitally modulated signals. First the signals are received by a conventional receiver and then digitized. A concentric finite impulse response (FIR) filterbank is used to band limit the digitized signal to N different bandwidths about the intermediate frequency of the receiver. The concentric FIR filterbank has N parallel outputs corresponding to the N different bandwidths.

The next stage of the processing is demodulation by what Liedtke calls a universal demodulator. "The name 'universal demodulator' indicates that all the modulation types of interest can be demodulated without specifically adjusting the demodulator parameters" (Liedtke,1984:313). The universal demodulator is realized by using many demodulators or by using only one demodulator in a time division multiplexed manner. The next step of the classification method is to calculate parameters of the unknown signal.

Feature extraction is the process of calculating attributes from input data (Tou and Gonzalez,1974:12). The features calculated by Liedtke are the amplitude, instantaneous frequency, and phase. The variances of the amplitude and instantaneous frequency data are calculated and histograms of the amplitude, instantaneous frequency and phase information are also computed.

The histograms are processed further by weighting functions. There is a specific weighting function for each modulation type of interest . Each weighting function has the property of producing a numerical result which is large when applied to the histogram from the type of modulation for which the weighting function is designed; the result is small when applied to histograms derived from other types of modulation. The next step in the classification method is to decide what type of modulation was used on the signal based upon the features which have been calculated.

Decision functions operate upon the processed features to decide which type of signal the features describe. The decision functions of Liedtke are based upon Boolean type equations. For example, if all of the following conditions are satisfied for the input data, BFSK is chosen as the type of modulation used on the input signal: the result of processing the frequency histogram of the data with the weighting function corresponding to BFSK is greater than the threshold for the processed frequency histogram; the variance of the instantaneous frequency is greater than its threshold; the variance of the amplitude is less than its threshold.

Liedtke uses the notation of Boolean algebra to simplify the expression of his decision functions. In his notation, the preceeding decision function is represented in equation (1-1) as

$$[FHI > TFHI] \text{ .AND. } [FVAR > TFVAR] \text{ .AND. } [AVAR < TAVAR] = TRUE \quad (1\text{-}1)$$

where

FHI	=	result of processing frequency histogram with the weighting function for BFSK
TFHI	=	threshold on processing frequency histogram with the weighting function for BFSK
FVAR	=	variance of the instantaneous frequency
TFVAR	=	threshold on the variance of the instantaneous frequency

AVAR	=	variance of the amplitude
TAVAR	=	threshold on the variance of the amplitude

Similar decision functions are given for the other modulation types of interest. Liedtke achieves good performance for the identification of the following types of modulation: BASK, BFSK, BPSK, QPSK, quaternary FSK and 8-PSK.

Summary of Jondral's Paper. The paper by Jondral describes a signal classification method very similar to that of Liedtke. The preprocessing of the signals are identical in both papers. The difference between the papers is that Jondral uses an adaptive process to develop his decision functions which are optimum in a mean squared error sense (Jondral,1985:184). Liedtke formulates his decision functions intuitively as boolean equations (Liedtke,1983). Jondral achieves good performance from his classification method for the following types of modulation: BASK, BFSK, BPSK2, quaternary FSK, amplitude modulation with large carrier (AM-LC) and single sideband amplitude modulation with suppressed carrier (SSB-SC).

Summary of Chan's Paper. The paper by Chan and others describes a method to determine the modulation type of a signal based upon the characteristics of its envelope. Note that this is just one of the features used by Liedtke and Jondral. However, the work of Chan and others show that the ratio of the variance of the envelope to the square of its mean can be used as a feature to reliably separate different types of modulation (Chan and others,1985). This ratio is derived as a function of carrier to noise ratio for the signals of interest and thresholds are calculated for the determination of modulation type. This scheme was shown to be effective for the separation of AM-LC, double sideband suppressed carrier AM, SSB, and FM. However, this method is unable to separate between classes of signals with constant envelopes. That is, it can not distinguish between classes of angle modulated signals since this type of modulation produces waveforms with constant envelopes (Chan and others,1985).

Assumptions

Several assumptions are made concerning the signals and the environment observed by the classification procedure presented in this thesis. The received signal is assumed to be corrupted by additive white gaussian noise. The signal which is to be processed is assumed to be at the output of the IF amplifier of a receiver. The IF is taken to be 100 kHz. It is also assumed that only the unknown signal plus noise is present within the passband of the IF amplifier. The message signal is assumed to have independent and equally likely symbols.

The assumptions mentioned above result in a mathematically tractable thesis problem which is readily implemented on a computer while simulating some of the conditions encountered in typical conditions.

Standards

The performance of the procedure developed in this thesis will be judged as successful or unsuccessful based upon the results of the simulation. Comparisons with the efforts of the work presented in the summary of current knowledge are inconclusive due to the limited number of samples classified by the developed method. However, results will be tabulated for the performance of the developed procedure versus signal to noise ratio and modulation type.

Approach

The approach to the signal classification problem is to simulate the signals and classification procedure in software. This method allows the precise control of the operating environment, signal and classifier parameters may be easily changed, and no specialized equipment is required.

The software is in Fortran and was written solely by the author with the exception of a fast Fourier transform routine, which is due to Ahmed and Natarajan (Ahmed and Natarajan,1983:160-161).

Comparison to Existing Methods. The approach to the classification problem will be a combination of the procedures of Liedtke, Chan and others, and Jondral. The method described in this thesis uses the mean and the variance of the signal envelope as two features. The decision functions used are developed from an adaptive algorithm. This is essentialy the same approach to the development of decision functions used by Jondral. The preprocessing for feature extraction is different from all of the above authors.

The methods of Liedtke and Jondral use phase histograms to determine the level of phase modulation of PSK signals, while the procedure due to Chan and others can not distinguish between classes of angle modulated signals. The automatic classifier described in this paper uses new methods to determine the level of modulation for PSK signals.

The original contributions of this effort are the application of new techniques for the separation of different levels of PSK signals. The separation of different levels of PSK refers to the determination of whether a phase shift keyed signal is BPSK or QPSK.

General Structure of Classification Procedure. The classification procedure consists of three steps. The first step is to calculate features from signals which are of known modulation type. The features are used as elements in a feature vector which are used as inputs to the next step. In the second step, these feature vectors are used as training vectors in an adaptive algorithm which produces weight vectors for each class of signals. After training, the third step is performed. Here, classification of unknown signals is performed by multiplying the weight vectors by the feature vector obtained from the unknown signal. The results of these multiplications are decision functions. These decision functions are such that the largest output occurs when a signal from the class for which it has been optimized is applied.

Feature Extraction. The features used in the classification method are derived from the envelope of the signal and from the spectra of the signal, the signal squared and the signal quadrupled.

The mean and variance of the envelope are calculated and used as elements of the feature vector. These features are intended to provide information necessary to classify amplitude shift keyed signals.

The magnitude of the Fourier transform of the signal is searched for energy of the chosen bandwidth using a correlation process which is described in the Theory chapter. The features obtained from this correlation are the magnitude and spectral location of two largest peaks of the correlation waveform. These elements of the feature vector are intended to provide infomation related to frequency shift keyed signals.

The magnitude of the Fourier transforms of the signal squared and the signal quadrupled are searched for narrowband energy near twice and four times the frequency obtained from the correlation of the spectrum of the original signal. The modulation from an M-ary PSK signal is removed when it is multiplied by itself M times (Proakis,1983:197). The result of this operation is an unmodulated sinusoid at M times the original carrier frequency.

Theoretically, the bandwidth of a sinusoid approaches zero as the observation time becomes infinite (Stremler,1982:87). In practice, the bandwith will be small, but zero bandwidth will not occur due to finite observation time and other effects. However, when a signal other than M-ary PSK is multiplied by itself M times, its bandwidth will be increased by a factor of M (Gagliardi,1978:63). The property of M-ary PSK signals producing a sinusoid when raised to the Mth power is exploited in this classification procedure. Since this property is unique to PSK signals, it is expected to be a useful feature for the separation of BPSK and QPSK from each other and other classes of signals.

Development of Decision Functions. The decision functions used in this experiment are generated by an adaptive technique known as the Least Mean Squares (LMS) algorithm (Widrow and Stearns,1985: Ch 6). It accepts feature vectors from known classes of signals. Based upon these inputs, the weights in an adaptive linear combiner, as shown in Figure 1-1, change so as to produce the largest value when the input signal is from the class to which the weights are matched.

The structure of the classifier of this paper uses an adaptive linear combiner for each class of signal. The class decision for an unknown feature vector is made by choosing the largest output from the set of adaptive linear combiners. The structure of the classifier is shown in Figure 1-2.

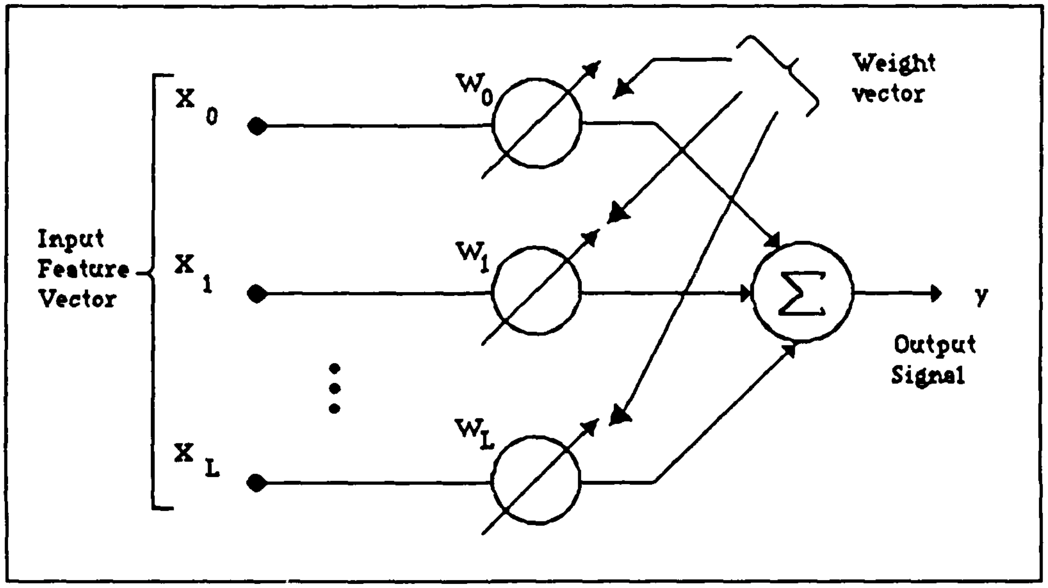


Figure 1-1. Adaptive Linear Combiner (Widrow and Stearns,1985:16)

Summary

This chapter has provided an overview of existing methods for the classification of the modulation type of signals. A brief presentation of the proposed method was also given. The existing methods are explored in greater depth in the next chapter and the proposed method is explained in the Theory chapter.

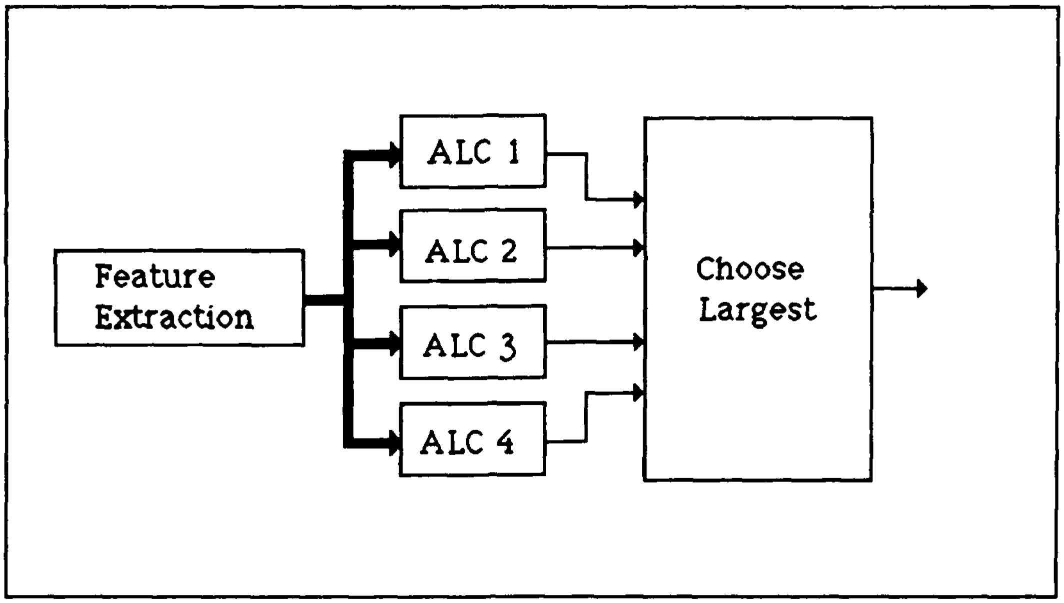


Figure 1-2. Structure of the Classifier

II. Literature Review

Background

The problem of identification of modulation type for digitally modulated signals is of interest in spectrum surveillance and electronic warfare applications. Communications jamming is one important aspect of electronic warfare. In the electronic warfare case, knowledge of the type of modulation used by an enemy emitter would allow an appropriate choice of a jamming signal (Golden,1983:12).

As stated in the first chapter, a review of the unclassified literature of the past five years resulted in the discovery of three papers concerned with the identification of the modulation type of unknown signals. The first paper to be considered is due to Liedtke (Liedtke,1984). The second paper examined is due to Jondral (Jondral,1985). Finally, the third paper is due to Chan and others (Chan and others,1985).

Liedtke's Classification Algorithm.

The earliest paper found was written by Liedtke in 1984. A computer simulation for the classification of signals according to modulation is described. The classes of signals considered for separation by the classifier are BASK, BPSK, QPSK4, 8-PSK, and BFSK.

Informational Relationships. An important aspect of electronic warfare is the jamming of communications signals. In this case, the jammer does not need to demodulate the underlying data of the enemy's signals. However, knowledge of the modulation type would assist the jammer in choosing a strategy (Golden,1983:12). The relationships between the amount of information required for signal detection, classification, and demodulation are considered below.

Figure 2-1 shows the amount of information gained after processingversus the amount of information required to perform the processing (Liedtke,1984:312). The figure shows that less a priori information is required for energy detection than for demodulation.

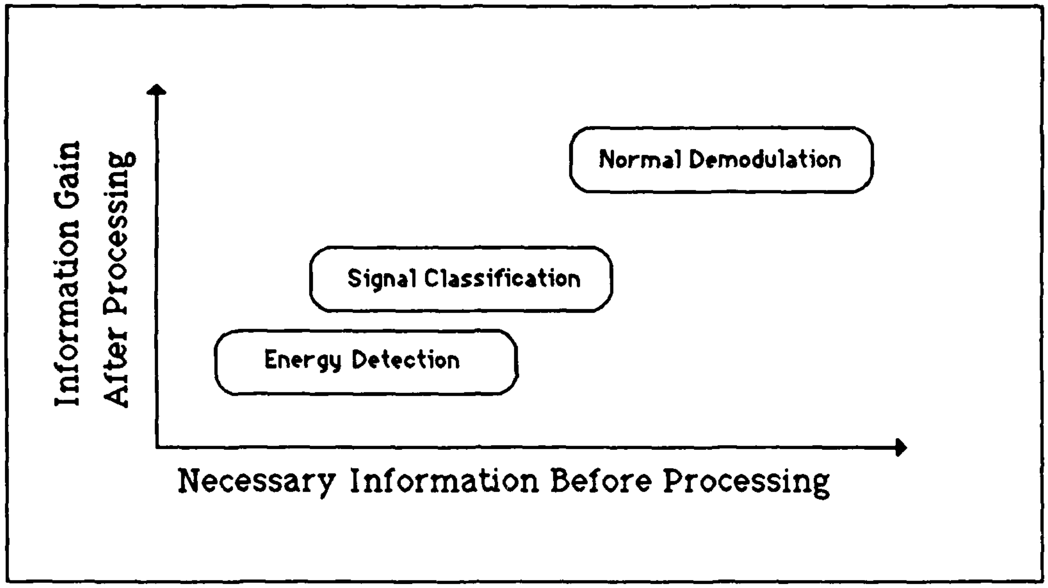


Figure 2-1. Informational Relationships (Liedtke,1984: 312)

However, the amount of information gained after demodulation is greater than that of energy detection. The informational relationships for signal classification are between the cases of demodulation and energy detection.

Energy detection requires the least amount of a priori information of the three processes considered in the figure. The center frequency of the unknown signal must be known only within a range determined by the bandwidth of the energy detector. However, energy detection provides only information related to the existence of radio frequency energy. Demodulation requires the largest amount of a priori information of the three processes shown in Figure 2-1. This information consists of modulation type, center frequency, bandwidth, symbol rate, and perhaps other parameters (Liedtke,1984:312). Correspondingly, demodulation recovers the most information from the signal of the three processes.

Classification requires less a priori information than needed for demodulation and more than needed for energy detection; the amount of information gained by classification is between the amounts from demodulation and energy detection. The structure of the classifier developed by Liedtke is discussed below.

Architecture of the Classifier. Figure 2-2 is a block diagram of the architecture of the classification system. The unknown signal enters the system through the antenna and receiver. The receiver is used only to translate a portion of the RF spectrum to the center frequency of components used later in the processing. However, this classification method requires an approximate value for the carrier frequency of the unknown signal. No demodulation occurs in the receiver.

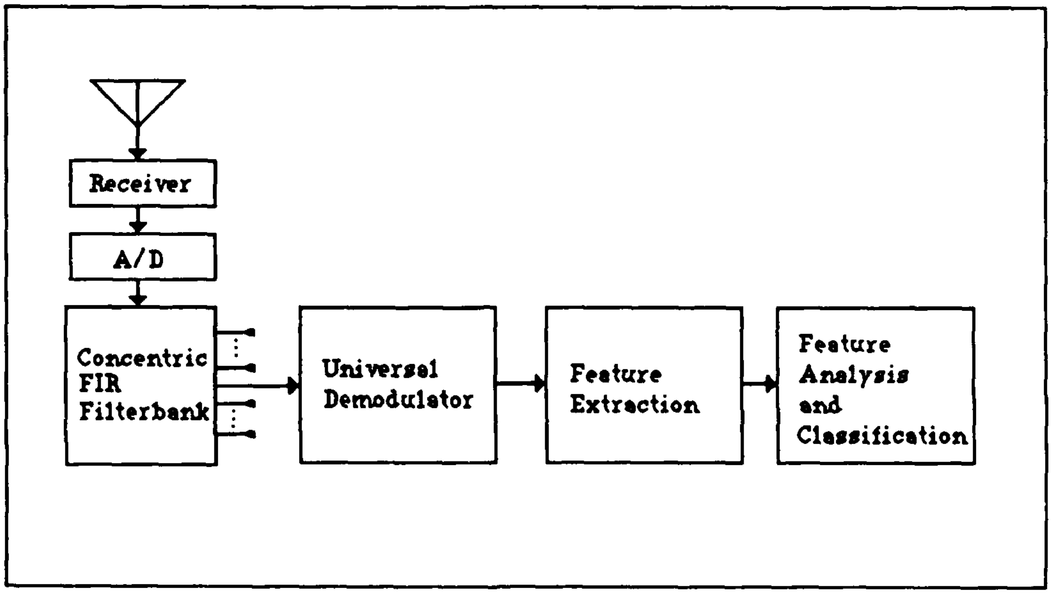


Figure 2-2. Architecture of Liedtke's Classification System (Liedtke,1984:313)

The output of the receiver is digitized and then filtered by a bank of FIR filters. The bank of FIR filters consists of a number of bandpass filters with the same center frequency but different bandwidths. The signal of interest is operated upon by all of the filters and then the filter outputs are processed individually. According to Liedtke, the best classification results are obtained from the output of the filter with the bandwidth that best matches the bandwidth of the unknown signal. This filter bandwidth also provides a measure of the keying rate of the signal. The outputs of the FIR filterbank are then input to a universal demodulator.

The universal demodulator of Liedtke is a demodulator which can demodulate all of the signals of interest without the adjustment of parameters. A bank of demodulators is suggested as a practical method of achieving the universal demodulator. Alternatively, one demodulator could be operated in a time division multiplexed mode under some form of automatic control (Liedtke,1984:313). Note that the signal has been digitized; therefore, demodulation is an algorithm implemented on a computer or special purpose digital hardware. The universal demodulator provides inputs to the feature extraction algorithms.

Feature Extraction. The feature extraction processing calculates parameters of the unknown signal that will assist in the classification of its modulation type. The features chosen by Liedtke are the amplitude, phase, and instantaneous frequency. The methods used to obtain these parameters are shown in Figure 2-3. The feature extraction process operates upon the digitized signal. Liedtke determines a sufficient sample rate by experiment. When the sample rate was eight times the bandwidth of the filter in the FIR filterbank, good classification results were obtained. The bandwidth of this filter is approximately equal to twice the reciprocal of the keying rate. Therefore, Liedtke was operating upon signals that were digitized at a rate which provided sixteen samples per symbol.

The feature extraction algorithm requires that the input signal be quadrature sampled. This is represented in Figure 2-3 by the real and imaginary inputs. The real and imaginary channels are also referred to as the inphase and quadrature components. The features that are calculated are functions of the inphase and quadrature components. The amplitude, phase,

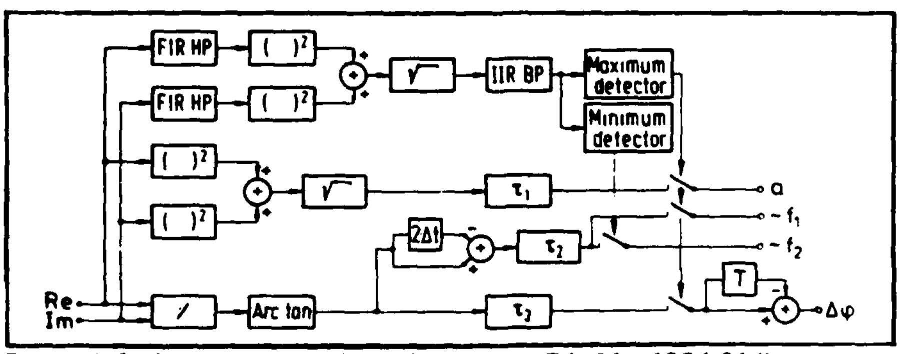


Figure 2-3. Feature Extraction Algorithms (Liedtke,1984:314)

and instantaneous frequency are calculated in a straightforward manner. Detailed explanations of these operations can be found in the references(Couch,1983;Schwartz,1980; Stremler,1982). Since the features are calculated at every sample instant, a method must be used to find the proper time to collect the ouptuts of the feature extraction.

Synchronization. The sampling instants for each feature are also calculated. This is not the same as the clock used for digitization. These sample times are used to determine when to extract the amplitude, phase, and frequency values from the feature extraction algorithm.

The correct times to collect the outputs of the feature extraction circuit are calculated by the upper signal path of Figure 2-3. Notice that outputs a, f_1, and $\Delta\phi$ are extracted based upon the maximum detector and that the output f_2 is extracted based upon the minimum detector. Working backwards along the signal path, it is seen that the inputs to the maximum and minimum detectors are the same signal. This signal is the square root of the sum of the squares of high pass filtered inphase and quadrature components. The purpose of the high pass filters is to remove the effects of modulation on the carrier.

That the amplitude and phase of the unknown signal should be measured at a maximum of the signal envelope is apparent. Also note that f_1 is extracted at a maximum. The phase differencing algorithm for the output f_2 is sampled at times determined by minima of the signal envelope.

Feature Processing. The features extracted by the previous step are used to generate histograms of the amplitude, frequency, and phase. This section describes the use of these histograms as related to the separation of BPSK, QPSK, 8-PSK, BFSK, and BASK.

The histograms generated from the phase values contain the phase difference between two sampled points as given by $\Delta\phi(kt) = \phi(kt) - \phi(kt - T)$ and the result is called the difference phase histogram. This was done because Liedtke has difficulty obtaining a correct reference phase (Liedtke,1984:315). The difference phase histograms of BPSK, and QPSK are shown in Figure 2-4.

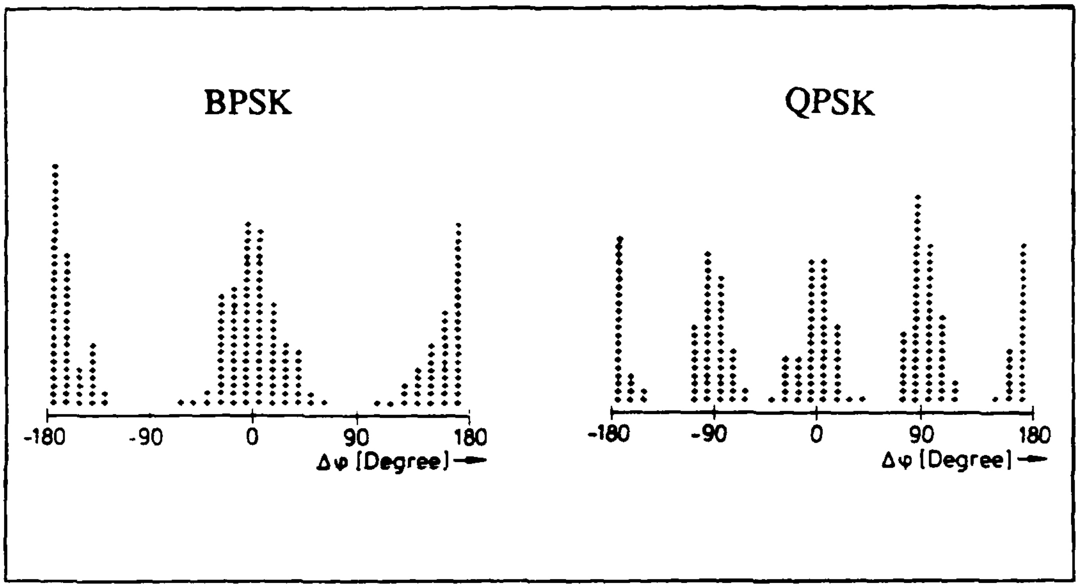


Figure 2-4. Difference Phase Histograms (Liedtke,1984:315)

The values have been modified such that all $\Delta\o$ values are between ± 180 degrees. The histogram of BPSK has peaks at 0 and ± 180 degrees. The three peaks of the histogram actually depict two phase states since a positive phase shift of 180 degrees is equivalent to a negative phase shift of180 degrees. Similarly, the histogram of QPSK has five peaks corresponding to the four phase states of this signal. The histogram for white gaussian noise does not have a structured appearance. These histograms are processed in such a way as to allow the separation of BPSK, QPSK, and PSK8.

The difference phase histograms are considered as waveforms to be processed. The object is to use the histograms as inputs to a procedure that produces a maximal output when the histograms are matched with the signal of interest. This structure can be viewed as a set of matched filters for an M-ary signaling set. Figure 2-5 shows a bank of matched filters used for optimum detection of M-ary signals. In the histogram separation problem, each histogram is considered as one signal of the M-ary signaling set. However, this implementation could not be used by Liedtke because the impulse response of the matched filters are not possible to calculate (Liedtke,1984:316). This is shown by consideration of a two class problem corresponding to only two possible classes of unknown signals.

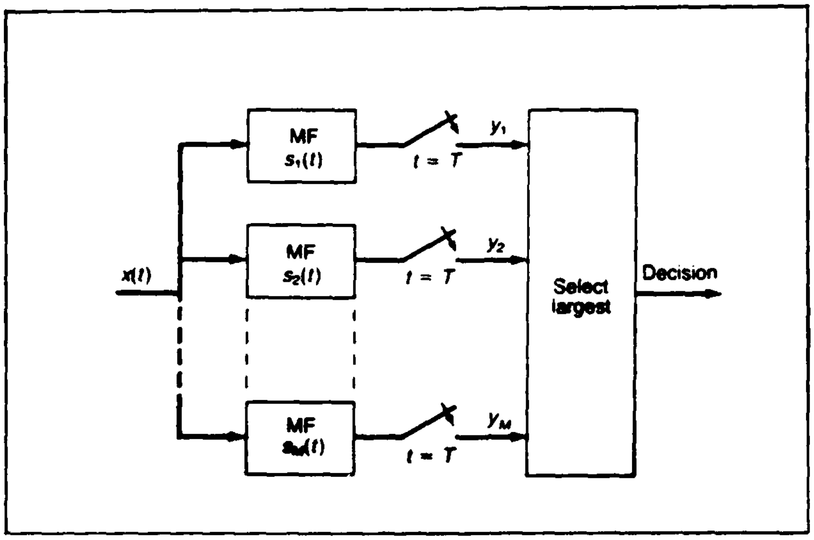


Figure 2-5. Matched Filter Processing for M-ary Signals (Cooper and McGillem,1986:221)

The likelihood ratio test involved in making a two class decision is given by Liedtke as (Liedtke,1984:316)

$$l(x_0, x_1, \ldots, x_{M-1}) = \frac{f(x_0, x_1, \ldots, x_{M-1} \mid C_1)}{f(x_0, x_1, \ldots, x_{M-1} \mid C_0)} > TL \qquad (2\text{-}1)$$

where

$f(x_0, x_1, \ldots, x_{M-1} \mid C_i)$ = conditional probability density function of histogram values given class i; $i = 0,1$

x_i = histogram values

M	=	number of cells in the histograms
TL	=	threshold chosen to optimize some condition

The problems associated with the computation of the conditional probabilty density functions are twofold; they are a function of the symbol energy to white noise energy density ratio and are also dependent upon the maximum value of the phase difference histogram which is a function of the message (Liedtke,1984:317).

These problems are overcome by the use of suboptimal weighting functions that produce maximal values when applied to the histogram for which they are matched. Weighting functions are developed for the signals considered in this paper. The weighting functions for BASK, BPSK, and QPSK are shown in Figure 2-6.

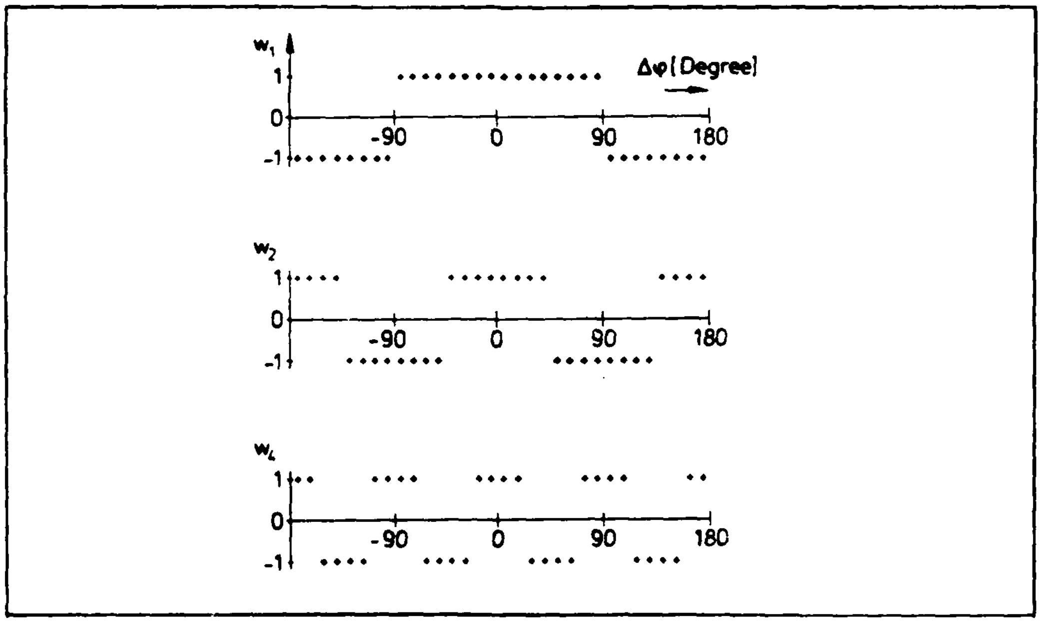


Figure 2-6. Weighting Functions for BASK, BPSK and QPSK (Liedtke,1984:316)

These weighting functions do not suffer from the same problems as the optimal weighting functions. Each weighting function has the property of producing a maximal value when it operates upon the phase difference histogram for which it is designed.

An example of feature processing for one class of signals is considered. Assuming PSK modulation was used on the unknown signal, the level of modulation is determined by operating upon the phase difference histogram by each weighting function and choosing the level of phase modulation corresponding to the weighting function which lead to the largest output. If two identical values are obtained from this process, the lower level of phase modulation should be chosen since a BPSK signal will have a phase difference histogram that will produce a large result when operated upon by the QPSK and 8-PSK weighting functions.

Actual signal classification is done by considering a series of two class problems. The first test separates BPSK, QPSK, and 8-PSK from noise by the approach described above. The result of this test also determines the level of phase modulation.

The second test is used in the separation of BPSK from BASK and BFSK. The variances of the amplitude and frequency (from maximum detector) values are calculated. Liedtke states that " a large amplitude variance value is indicative of BASK, and a large frequency variance is indicative of BFSK." BPSK would have small values for both amplitude and frequency variance.

A third test is used to separate BASK and BFSK from noise. It is similar to the test for separating PSK from noise. The amplitude histogram of BASK contains two peaks as does the frequency histogram of BFSK. These histograms will contain only one peak for other types of modulation (Liedtke,1984:317).

A review of the classification procedure reveals the five features used in the automatic classification method. These features are the difference phase histogram, the amplitude histogram, the frequency histogram (with the frequency values determined at a minimum sampling instant), the amplitude variance, and the frequency variance (with the frequency values determined at a maximum sampling instant).

Decision Functions. The five separation parameters defined above are used in decision functions to perform the classification of unknown signals. Liedtke uses Boolean type equations to specify decision functions. The decision function for PSK with i phase states is by Liedtke as (Liedtke,1984:318)

$$[(\max_{w_i}(\mathrm{DPHI}))_{i>1} > \mathrm{TDPHI}] \cdot [\mathrm{AVAR} < \mathrm{TLAVAR}] \cdot [\mathrm{FVAR} < \mathrm{TFVAR}] = \mathrm{TRUE} \qquad (2\text{-}2)$$

where

$(\max_{w_i}(\mathrm{DPHI}))$	=	selecting the largest value resulting from the processing the difference phase histogram with the weighting functions w_1, w_2, w_4, and w_8
DPHI	=	result of processing a phase difference histogram with a weighting function
TDPHI	=	threshold of the phase difference histogram
AVAR	=	amplitude variance
TLAVAR	=	lower threshold of amplitude variance
FVAR	=	frequency variance
TFVAR	=	threshold of frequency variance

The dots between the square brackets symbolize the logical "AND" operation. Each expression in brackets is evaluated as a logical binary decision. Then each bracketed term is logically AND'ed and the result is compared to the right hand side of the expression. This expression is interpreted as: choose PSK with i phase states if the result of processing the difference phase histogram with weighting function i is greater than any other weighting function j ($i \neq j$) and the amplitude variance is less than a lower threshold of the amplitude variance, and if the frequency variance is less than a threshold of the frequency variance.

Liedtke's decision functions for BASK and BFSK are given in equations (2-3) and (2-4) as

$$[\mathrm{AHI} > \mathrm{TAHI}] \cdot [\mathrm{AVAR} > \mathrm{TUAVAR}] = \mathrm{TRUE} \qquad (2\text{-}3)$$

$$[FHI > TFHI] \cdot [FVAR > TFVAR] \cdot [AVAR < TLAVAR] = TRUE \quad (2\text{-}4)$$

where

AHI	=	result of processing amplitude histogram with w_1
TAHI	=	threshold for AHI
TUAVAR	=	upper threshold of amplitude variance
FHI	=	result of processing frequency histogram with w_2
TFHI	=	threshold for FHI

A conceptualized decision space is shown in Figure 2-7. The dotted lines represent thresholds. Threshold values were chosen more than three standard deviations away from the mean values of the separation parameters. The arrows indicate the directions of increasing feature values. The results of the simulation are presented in the next section.

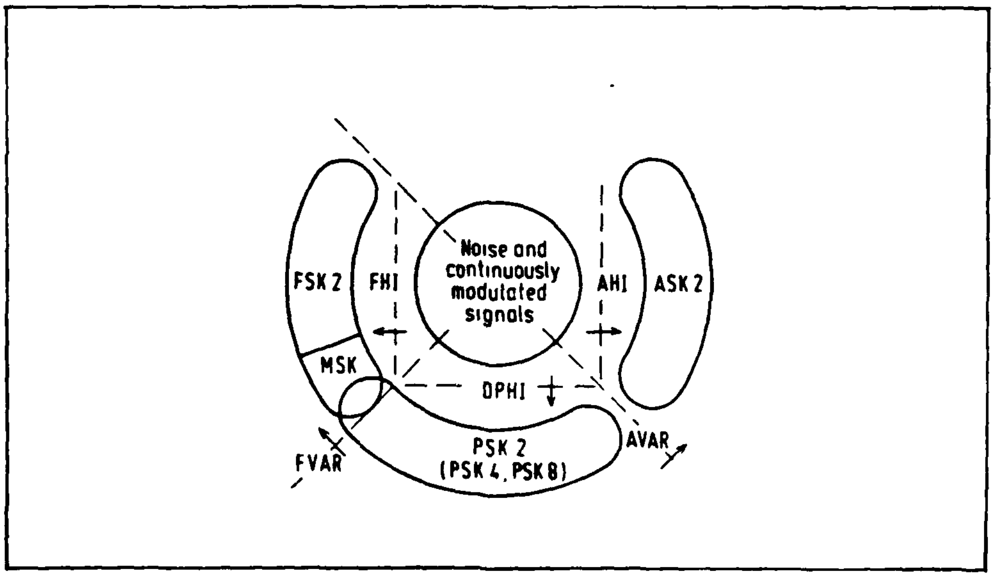


Figure 2-7. Conceptualized Decision Space (Liedtke,1984:318)

Results. The ability of the classification method to discriminate against noise was tested by running 100 simulations with white gaussian noise as the only input. The classifier never misidentified noise as a type of digital modulation. Figure 2-8 presents the results of the classifier on the signals of interest. The probability of a correct decision by the classifier is represented by the symbol P_d. The values of P_d where estimated by running a 256 symbol length message through the simulation 25 times for each E_T/n_0 value plotted.

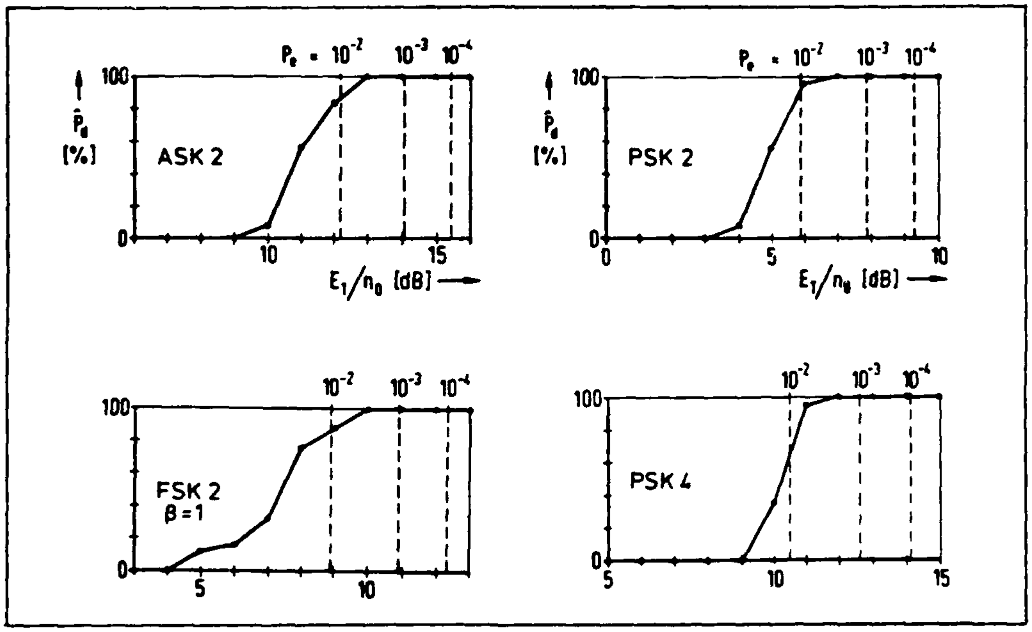


Figure 2-8. Results of Classification (Liedtke,1984:319)

This method was also shown to perform well under conditions of practical interest; the clasifier was tested for its ability to separate signals when the center frequency of the unknown signal was mistuned, the symbol rate was not estimated properly, and the signal was located between two frequency channels of similar signal strength and modulation type.

Summary. This classification algorithm has been shown to perform well at signal to noise ratios that are likely to be encounterd in practical situations. Liedtke presents graphs of the probability of correct classificaton versus E_T/n_0 .

Jondral's Classification Algorithm.

The second paper of the literature review was written by Jondral in 1985 (Jondral,1985). The structure of the classification algorithm used by Jondral is shown in Figure 2-9. The experiment considers the following seven types of signals: BASK, BFSK, QFSK, BPSK, AM, SSB-SC, and noise. Jondral refers to AM, and SSB-SC as A3 and A3J.

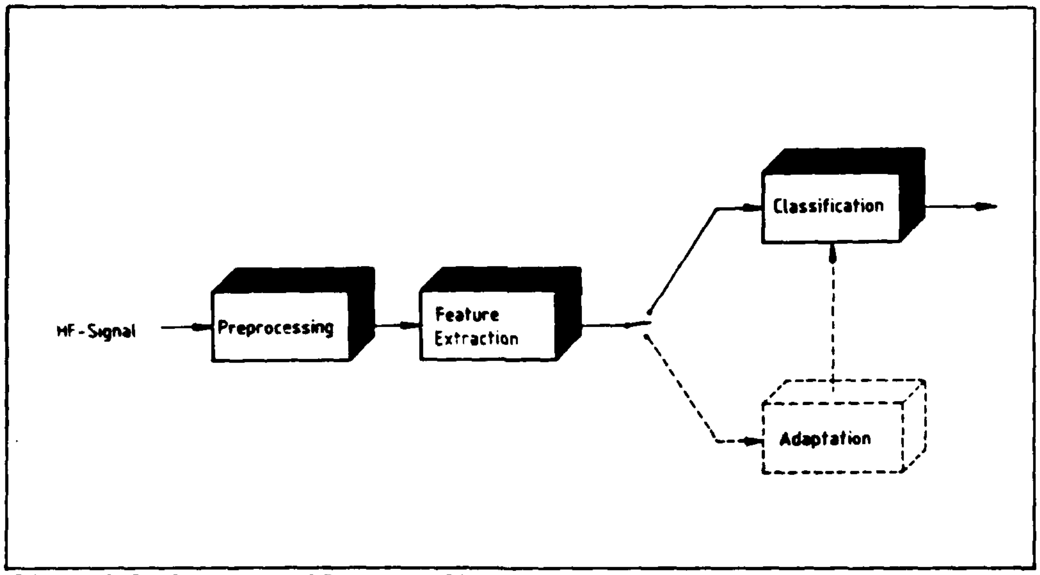


Figure 2-9. Structure of Jondral's Classifier (Jondral,1985:178)

The preprocessing stage is functionally identical to the preprocessing of Liedtke. Another similarity to the classifier of Liedtke is that the features used in this classifier are derived from normalized histograms of the amplitude, phase, and frequency of the signal (Jondral.1985:182).

The similarities with Liedtke's paper end with the classification procedure. Although the features generated by Jondral are histograms, the values from each histogram are then concatenated with each other to form a vector of 192 elements. Feature vectors for the signals of interest to Jondral are shown in Figure 2-10. The classification of unknown signals is based upon the ability of decision functions to distinguish between these feature vectors.

Classification. Jondral uses a two step classification process. The first step of the classification procedure uses signals from known classes. Feature vectors are calculated from these signals. These feature vectors are then used to train an adaptive classifier. The adaptation of the classifier results in coefficient vectors. The result of multiplying weight vectors with feature vectors are known as decision functions. The decision functions are shown to be weighted sums of the elements of the feature vectors (Jondral,1985: 184). The adaptation process results in weight vectors which minimize the mean squared error between the desired and actual outputs (Jondral, 1985: 184).

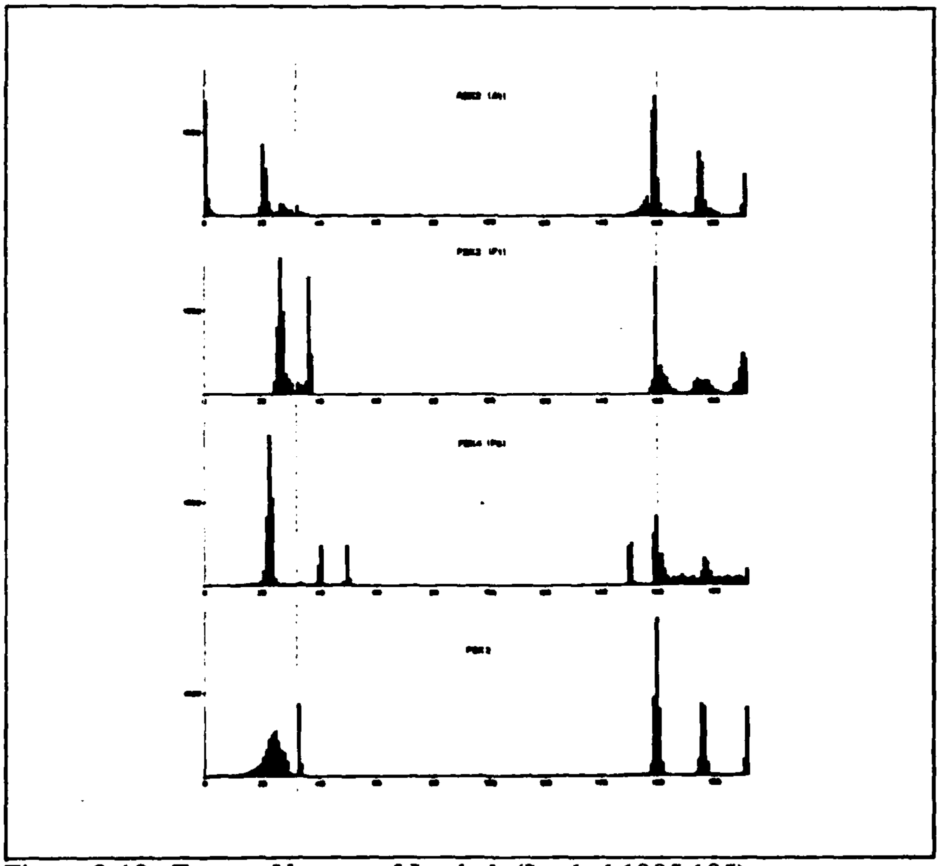


Figure 2-10. Feature Vectors of Jondral (Jondral,1985:185)

The second step in the classification process is to use the coefficient vectors together as a matrix to multiply with feature vectors from unknown signals. The result of this multiplication is a column vector. The elements of this column vector correspond to classes of signals. For example, if the third element in the resultant column vector is the largest of all the elements, the classifier of Jondral decides that the unknown signal belongs to class 3 (Jondral,1985:184).

Experimental Results. The unknown signals used in this experiment were not simulated in software. Radio signals were recorded on magnetic tape under the supervision of a listener who classified the type of modulation used on each signal. The classification given by the listener is taken to be the actual modulation used on the signal. Therefore, the result of the automatic classifier is considered correct when it is same conclusion as the human classifier (Jondral,1985:186).

The adaptation of the classifier was done on a set of learning samples. Classification was then performed on other samples to determine how well the classifier performed. The number of learning samples during the adaptation for each signal of interest is shown in Table 2-1. After learning was completed, the classifier was used on the test signals. The results are presented in Table 2-2.

Summary. Jondral's approach to signal classification uses essentially the same features as Liedtke. However, Jondral uses an adaptive process to form weight vectors for use in the pattern recognition algorithm. However, the results of the two papers can not be directly compared because Jondral does not include performance as a function of signal to noise ratio. Signal to noise ratios of the signals used in the classification procedure are not known. However, all SNR's were sufficient to allow a human to perform visual or aural classification. Without knowledge of the SNR's, quantitative performance comparisons between this classification technique and others can not be made.

Chan's Classification Algorithm.

The third paper of the literature review was written by Chan, Gadbois, and Yansouni in1985 (Chan and others,1985). A method is presented for the identification of the modulation type of an unknown signal based upon the statistics of its envelope. The ratio of the variance of the envelope to the square of its mean is used as the only feature in this signal classification scheme.

Background. The feature used for separation, the ratio of the variance of the signal envelope to the square of the mean of the signal envelope, is known as R. The use of R for modulation identification can be understood at an intuitive level by considering a frequency modulated signal. In

Table 2-1. Number of Learning and Test Samples for each Signal

Signal Class	Learning Samples	Test Samples
BASK	772	257
BFSK	1256	418
QFSK	1109	370
BPSK	1500	500
AM-LC	1500	500
AM-SSB-SC	916	306
Noise	1500	500
Sum	8553	2851

(Jondral,1985:187)

Table 2-2 Classification Results after Learning

	BASK	BFSK	QFSK	BPSK	A3	A3J	Noise
BASK	91.8	1.2	0.0	1.9	0.4	3.1	1.6
BFSK	0.0	95.2	0.2	1.2	0.0	1.0	2.4
QFSK	0.0	4.3	88.1	0.0	0.0	3.3	4.3
BPSK	0.0	0.0	0.0	95.8	1.8	0.0	2.4
A3	0.0	0 .2	0.0	2.4	95.4	0.0	2.0
A3J	3.3	0.3	1.0	0.3	0.3	83.3	11.5
Noise	0.0	0.2	0.0	0.0	0.0	4.0	95.8

(Jondral,1985:188)

frequency modulation, the information is contained in the instantaneous frequency of the signal: an FM signal has a constant envelope (Stremler,1982: 279). The variance of its envelope is zero and therefore, R is equal to zero. For amplitude modulation, the information is conveyed by the envelope. Chan and others show that R approaches unity for AM.

Through the use of similar intuitive arguments, this method can be shown to be unable to separate constant envelope signals such as FM, FSK, and PSK. However, the following types of amplitude modulation, SSB, DSB-SC, and DSB-LC, have been shown to "have very distinctive" R values (Chan and others).

Architecture of Chan's Classifier. A conceptual diagram of the modulation identification method is shown in Figure 2-11. Assuming a quadrature sampled signalas in the methods of Liedtke and Jondral, the envelope is calculated. The feature processing then consists of calculating the variance and mean squared value of the envelope. The ratio of the variance to the square of the mean is then calculated. The decision function is a thresholding operation which classifies signals based upon the value of the ratio calculated previously.

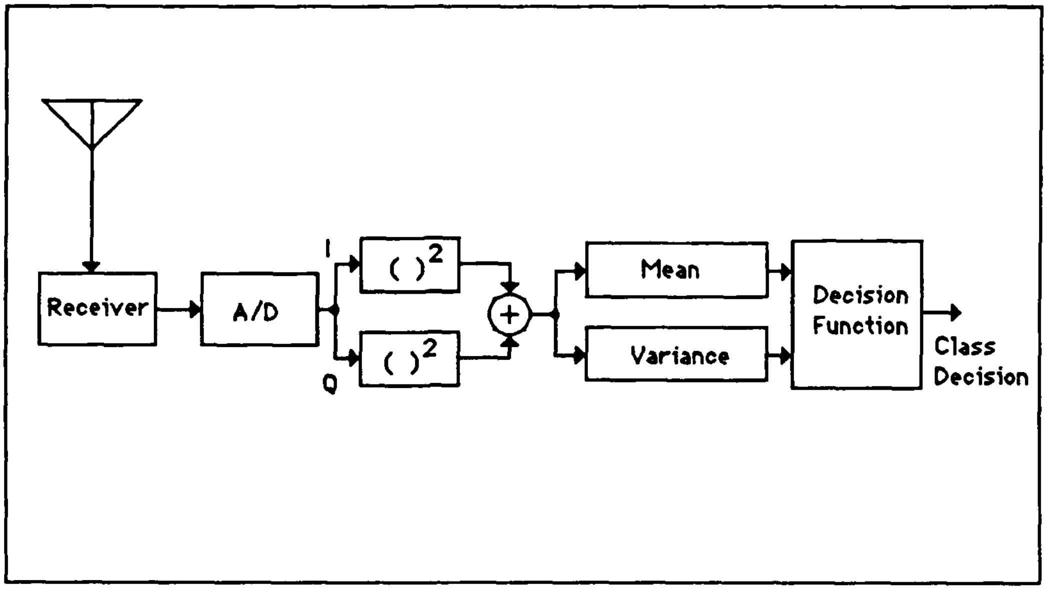


Figure 2-11. Architecture of Chan's Classifier

Decision Functions. Chan and others calculate theoretical values of R for the modulation types listed in the Background section. These theoretical values are compared to experimentally obtained values from 200 trials at two carrier to noise ratios and are displayed in Table 2-3. The experimental and theoretical values are within close agreement. The decision rules are based upon the theoretically obtained values for R are shown in Table 2-4.

The experimental data was generated with a gaussian message, gaussian noise, and 2048 points of bandpass signal centered at 40 kHz and sampled at 160 kHz. Table V shows the results of this classification for 200 trials of the experiment at a carrier to noise ratio of 7dB.

Summary. This classification procedure has been shown to operate well at a carrier to noise ratio of 7dB. This is below the threshold for FM communication (Gagliardi,1978: 159). However, Table 2-5 shows that during 200 simulations FM was never mistaken for a different type of modulation. This reliable separation of constant envelope signals from varying envelope signals at the expense of not being able to distinguish between the classes of constant envelope signals.

Table 2-3. Experimental and Theoretical Values of R

Type	CNR	R_{exp}	R_{the}	R_{exp}	R_{the}
FM	7.0	0.31	0.31	0.019	0.012
	10.4	0.16	0.16	0.0099	0.0057
AM	7.0	0.79	0.79	0.073	0.040
	10.4	0.76	0.76	0.076	0.038
SSB	7.0	1.00	1.00	0.080	0.054
	10.4	1.00	1.00	0.097	0.054
DSB	7.0	1.31	1.31	0.14	0.077
	10.4	1.54	1.54	0.20	0.097

(Chan and others,1985:22.5.4)

Table 2-4. Decision Rule

R	Decision
$0.396 \geq R$	FM
$.897 \geq R > .396$	AM
$1.105 \geq R \geq .897$	SSB
$R > 1.105$	DSB

(Chan and others,1985:22.5.4)

Assuming that the AM signal could be modulated by an antipodal $\pm$ 1 bit stream, this technique can be compared quantitatively to Liedtke's technique. The above signal is identical in form to a BPSK signal. Liedtke obtains a probability of detection of unity for BPSK at a CNR of 7 dB while Chan and others have a probability of detection of 0.91at a CNR of 7dB (Liedtke,1984: 319; Chan and others,1985:841).

Table 2-5. Classification Results

	FM	AM	SSB	DSB
FM	200	0	0	0
AM	0	181	19	0
SSB	0	15	160	25
DSB	0	0	12	188

(Chan and others,1985:22.5.4)

The complexity of Liedtke's procedure provides better performance than the simpler method of Chan and others. However, less processing is required for the latter method. The theory supporting the classification procedure of this paper is presented in the next chapter.

III. Theory.

Introduction.

This chapter presents the theory used in the development and implementation of the signal classification procedure. The classifier developed here contains some of the elements from the three papers of the literature review and some new features which will be discussed in later sections.

The architecture of the classifier is shown below in Figure 3-1.

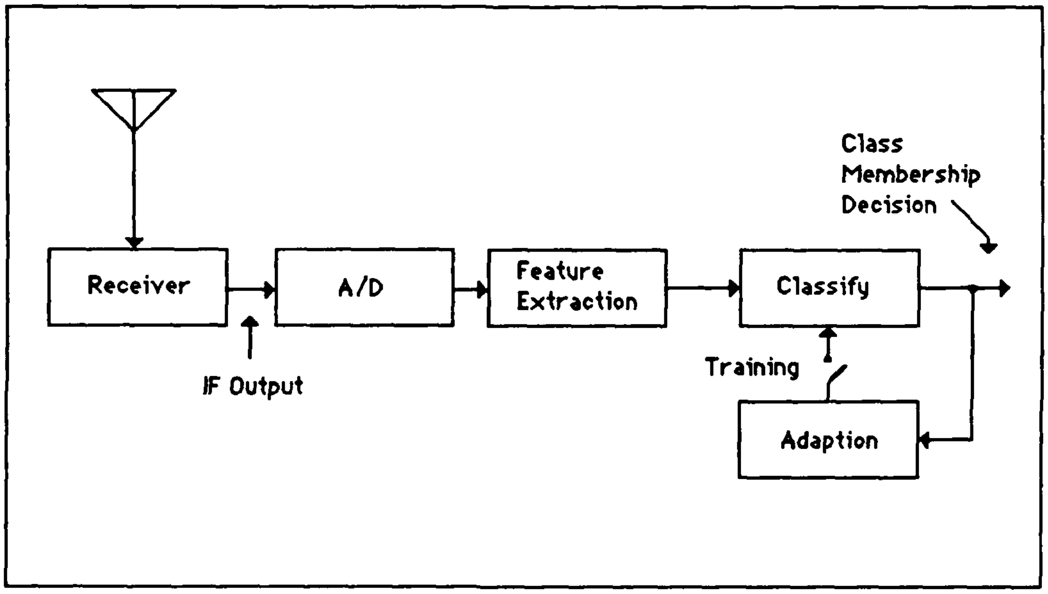


Figure 3-1. Architecture of the Classifier

The objective of the classifier is to determine the modulation type of the unknown signal. The classification procedure is based upon building vectors whose elements are features calculated from the signal. These vectors are considered as patterns and are input to a set of linear decision functions generated by an adaptive algorithm. The feature extraction and pattern classification procedures are now described.

Description of Features.

The features used in the classification of the modulation types for digitally modulated signals are presented in this section. The first two features are derived from the envelope of the signal and the following features are obtained from spectra related to the signal.

Features from the Signal Envelope. The mean and variance of the envelope of the signal are calculated and are used as two elements of the feature vector. The remaining features are derived from the spectrum of the signal and spectra of waveforms related to the signal.

Features from the Signal Spectra. A spectral correlation technique is used for the extraction of the remaining features. The concept of spectral correlation is discussed below.

Spectral Correlation. Correlation is a mathematical technique which is used to determine the similiarities between functions. This technique is routinely used with time domain signals. The approach used in this thesis is to search the spectra of unknown signals for a common feature using correlation.

The spectral feature common to all digital modulation schemes considered in this paper is that their energy is distributed in a $\mathrm{sinc}^2(x)$ manner about the carrier frequency. The sinc function is defined by Couch as $\mathrm{sinc}(x) = \sin(\pi x)/\pi x)$ (Couch, 1983:20). Therefore, a correlation of the spectra with a $\mathrm{sinc}^2(x)$ function will result in a peak when the shift equals the carrier frequency. However, the widths of the spectral lobes are functions of the symbol rate of the modulation. This experiment simulates only signals with a symbol rate of 2500 symbols per second. Other symbol rates could be accomodated by reference functions of different bandwidths.

Four elements of the feature vector are calculated as follows. First, the spectrum of the unknown signal is correlated with a $\mathrm{sinc}^2(x)$ function whose bandwidth is 5 kHz. This corresponds to signals with a symbol rate of 2500 symbols per second. Then, the results of this correlation is searched for the largest two values. The maginitude of the peaks and their spectral locations are saved as features.

Next, features are calculated from the spectrum of the signal squared. This spectrum is correlated with a narrowband $\mathrm{sinc}^2(x)$ function. The bandwidth of this $\mathrm{sinc}^2(x)$ function will be determined empirically. The resultant waveform is searched for a peak in the region of twice the intermediate frequency. The magnitude and spectral location of the largest peak constitutes two more elements of the feature vector. A similiar procedure is used to obtain the next two elements of the feature vector.

The spectrum of the quadrupled signal is correlated with a narrowband $\mathrm{sinc}^2(x)$ function. The resultant waveform is searched for a peak near four times the intermediate frequency. The largest peak of this correlation and its location are used as the following two elements in the feature vector. Explanations for the extraction of the above features are given in the next section.

Physical Significance of Elements in the Feature Vector

This section presents an intuitive explanation of the significance of the elements used to form the feature vector. First, the features derived from the envelope of the signal are discussed.

Features from the Signal Envelope. The mean and variance of the signal envelope are the first two elements of the feature vector. The mean of the envelope is its average value while its variance is a measure of the concentration of envelope values about the mean (Ziemer and Tranter, 1976: 292). The envelope variance of constant envelope signals such as M-ary PSK and M-ary FSK is theoretically zero (Chan and others,1985:22.5.2). The variance must be other than zero if information is conveyed by the envelope, such as in any form of AM.

In the previous chapter, Chan and others have shown for certain modulation types that the ratio of the variance of the envelope to the square of its mean can be used to classify the modulation type of certain unknown signals. The division of the variance by the square of the mean serves as a normalizing procedure. This normalization provides a relative measure of changes in the envelope with respect to its average value.

Features from the Signal Spectra. Estimates of the carrier frequency, or frequencies as in BFSK, are obtained from the spectrum of the signal. Features useful for identification of BPSK and QPSK are obtained from the spectra of the signal squared and quadrupled.

When the spectrum of the signal is correlated with the $\text{sinc}^2(x)$ function a waveform is produced. The two largest peaks and their locations from the resultant waveform provide the next four elements of the feature vector. The purpose of these features are to provide information related to the carrier frequency, or frequencies, of the unknown signal. The estimate of the carrier frequency is used in the following step and as a feature to indicate BFSK. An example is presented to illustrate these principles.

The theoretical power spectral density (PSD) of a BASK signal is shown in Figure 3-2. The width of the main lobe is twice the keying rate and the main lobe is centered about

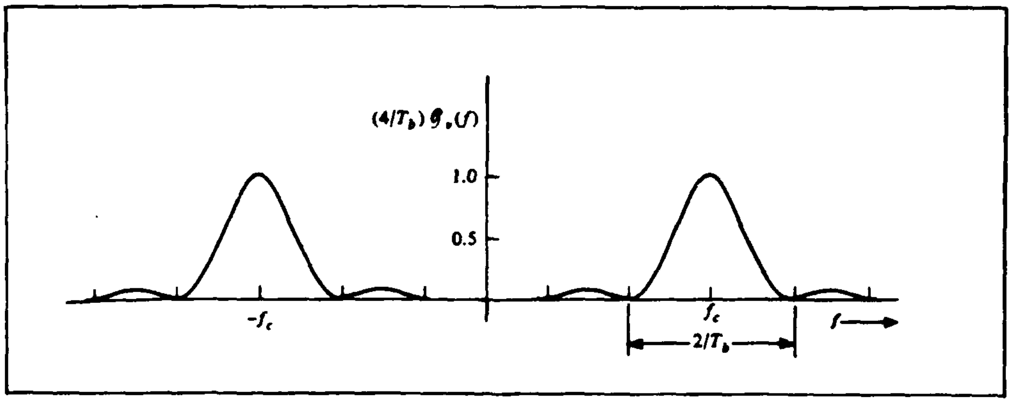


Figure 3-2. Theoretical PSD of BASK (Couch,1983:35)

the carrier frequency (Schwartz, 1980: 215). This spectrum is treated as a waveform in the following procedure. That is, a technique commonly used in the time domain will be used in the frequency domain. The procedure is the same as in a time domain correlation. The only difference is that the delay variable in the spectral correlation represents a frequency shift instead of a time shift.

An optimum method for locating the $\mathrm{sinc}^2(x)$ shapes in the signal spectra is desired. The processor that maximizes the peak signal to noise power ratio of a pulse in gaussian noise is the matched filter (Cooper and McGillem, 1986: 88). In this case, the signal shape is the $\mathrm{sinc}^2(x)$ function in the spectra of the signal. The matched filtering is accomplished using correlation which is equivalent to matched filtering under certain conditions (Cooper and McGillem, 1986: 90). This equivalence is shown in Figure 3-3.

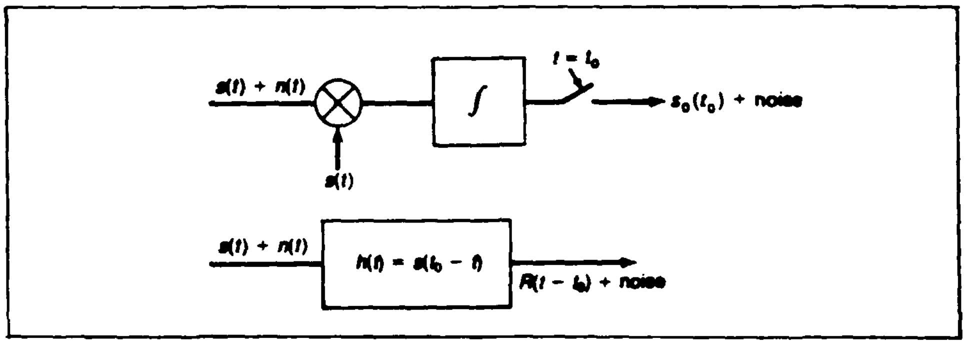


Figure 3-3. Equivalence of Matched Filter and Correlator (Cooper and McGillem,1986:90)

Recall that the objective of this portion of the feature extraction is to determine the center frequency of the unknown signal. Therefore, the spectrum of the signal, in this case the BASK spectrum of Figure 3-2, is correlated with the reference function of the form $\mathrm{sinc}^2(x)$. The baseband $\mathrm{sinc}^2(x)$ is shown in Figure 3-4.

The maximum value of the correlation will occur when the reference function is shifted such that it is aligned with the center frequency of the BASK signal. The amount of frequency shift to the peak of the correlation provides an estimate of the carrier frequency. The result of correlating the PSD of Figure 3-2 with the baseband $\mathrm{sinc}^2(x)$ of Figure 3-4 is shown in Figure 3-5.

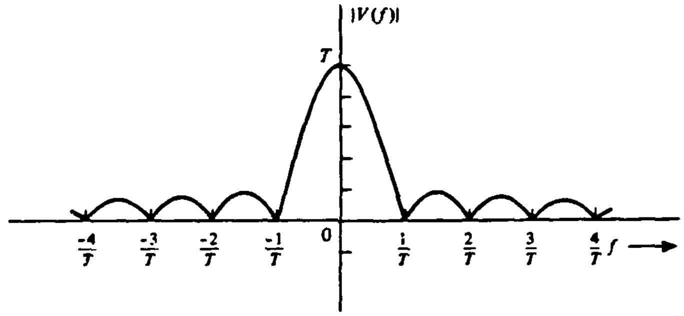


Figure 3-4. Baseband sinc^2(x) Function used for Correlation (Couch,1983:23)

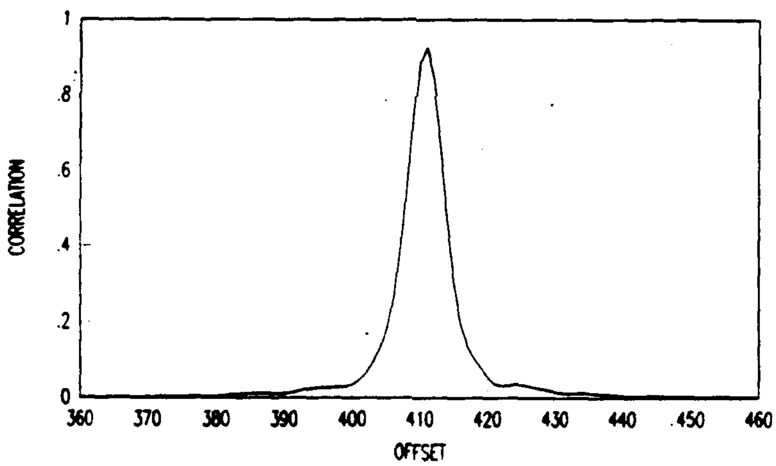


Figure 3-5. Result of Correlation of PSD of BASK with Baseband sinc^2(x)

The two largest values of the results of the correlation of the reference function with the spectrum of the unknown signal are saved to provide information related to FSK signaling. The PSD of BFSK signals is shown in Figure 3-6. The figure assumes frequency spacing which relults in orthogonal symbol waveforms. The result of the correlation of the sinc^2(x) with BFSK will contain two peaks due to the two peaks of the

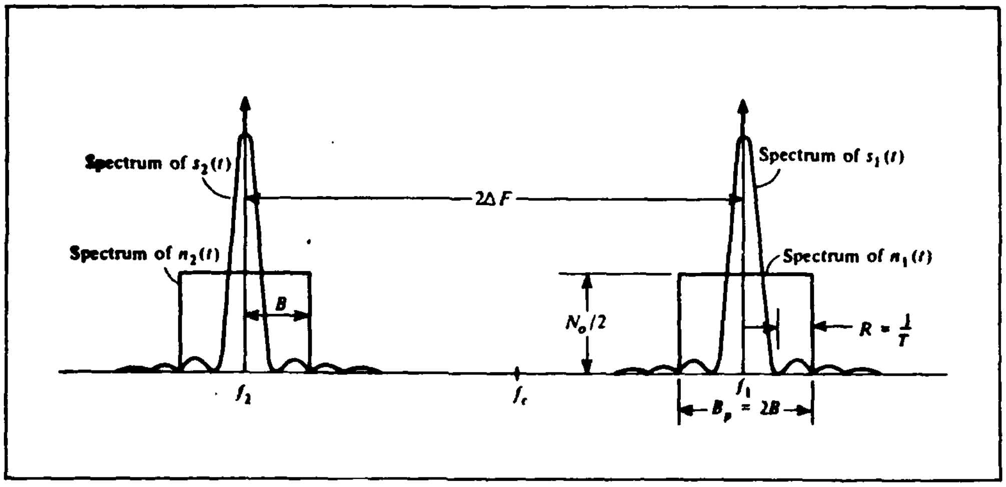


Figure 3-6. Spectrum of a BFSK Signal (Couch,1983:356)

spectrum. This is the purpose of retaining more than just one set of peak and location values from the correlation.

Features from Spectra of Signal Raised to Powers. The preceeding steps have resulted in features that assist in the detemination of carrier frequency or frequencies. The remaining features to be calculated assist in the determination of the number of phase states for phase shift keyed signals.

The following two features are based upon an idea related to carrier recovery for M-ary PSK signals. A carrier recovery circuit for BPSK signals is shown in Figure 3-7. The first step in the process is to raise the signal to the second power. This results in a sinusoid at twice the carrier frequency of the input signal (Proakis, 1983:193). A bandpass filter tuned to this frequency is used to separate other unwanted spectral components. Then a frequency divider is used to provide a coherent reference signal at the carrier frequency (Proakis,1983:193).

The property exploited in this feature extraction process is that BPSK signals squared theoretically result in a sinusoid at twice the carrier frequency while others signals will have approximately twice the bandwidth of the original signal (Gagliardi, 1978:63). Therefore, a

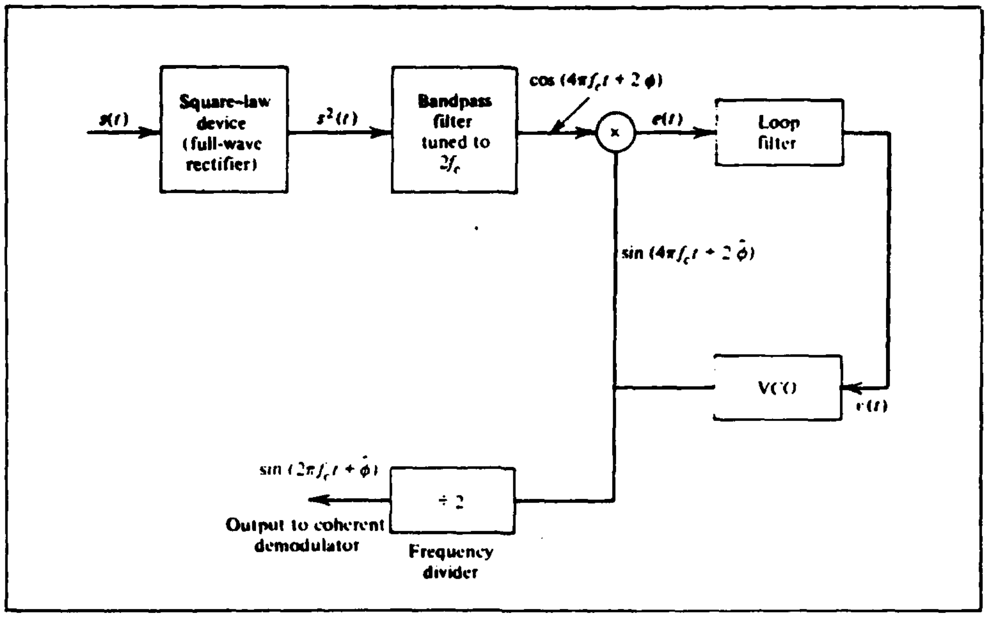


Figure 3-7. Carrier Recovery Circuit for BPSK (Proakis,1983:194)

narrow spectral peak at twice the carrier frequency is searched for using the spectral correlation technique. The presence of narrowband energy at twice the carrier frequency is a feature indicative of BPSK signals.

A similiar approach is used for QPSK except that the input signal is raised to the fourth power. Then, the correlation technique is used to search for narrowband energy at four times the carrier frequency.

In the discussion concerning raising the signal to the second and fourth powers, it has been assumed that there is sufficient signal power at the outputs of the nonlinear devices to obtain useful features. Analyses of square law devices in the presence of noise are presented in many texts (Cooper and McGillem,1986:118; Ziemer and Tranter,1976:270 ; Taub and Schilling,1986: 363). However, an analysis of the signal to noise ratio relationships of a fourth law devices is not as easily found. Appendix A provides such an analysis. The result is similiar to that of a square law device in that there is a threshold effect at an input signal to noise ratio of about 10 dB. Therefore, useful output is expected when the input SNR is above 10 dB.

This completes the discussion of the significance of the elements of the feature vector. Each element has been shown to be related to some unique aspect of an unknown signal. The remaining step in the process is the classification algorithm to operate upon the feature vectors.

Description of Classification Algorithm

The method used for classification involves two stages. In the first stage, weight vectors are generated from feature vectors calculated from signals with known class membership. The LMS algorithm is used to adaptively calculate the four weight vectors needed for the separation of the four classes of interest. The second stage, classification of unknown signals, begins after the weight vectors are calculated. Feature vectors from unknown signals are multiplied with the weight vectors. Class membership is determined by selecting the class corresponding to the weight vector which produces the largest ouput.

The LMS algorithm can be derived from a simpler algorithm, the perceptron algorithm (Lippmann,1987: 14). Therefore, the perceptron algorithm is described and then, the conversion from the perceptron to the LMS algorithm is presented.

The perceptron algorithm is an adaptive procedure whereby the algorithm modifies weight vectors to achieve optimum performance based upon the criterion of correctly identifying all the feature vectors of the training set (Tou and Gonzalez, 1974:162).

The adaptation is also referred to as training of the classifier. The training requires that known inputs be applied in order that the desired outputs are known. The training is considered complete when the algorithm no longer changes the elements of the weight vectors. The result of the perceptron algorithm are weight vectors which are used to form linear combinations of the elements in the feature vectors. The perceptron algorithm is now discussed in greater detail.

Perceptron Algorithm. Figure 3-8 shows a model of the perceptron classifier. The S array represents the elements of the feature vector. The A array represents associative units which perform a type of threshold logic.The perceptron algorithm uses a hard limiter as the function of the associative unit. Other possible functions for use in the associative units are given by Lippmann and are shown in Figure 3-9. Different versions of the perceptron are acheived by choosing different functions in the associative units. The LMS algorithm may be obtained from the perceptron algorithm by a substitution of the threshold logic function of Figure 3-9 for the hard limiter function (Lippmann, 1987:14).

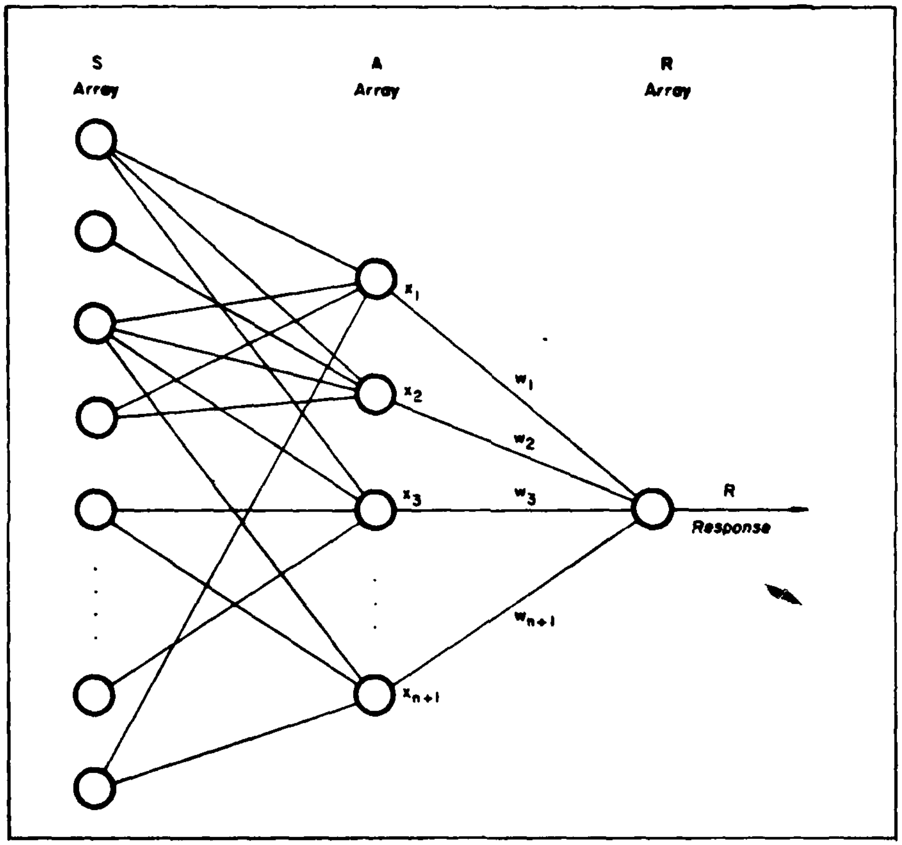


Figure 3-8. Basic Perceptron Structure (Tou and Gonzalez, 1974:160)

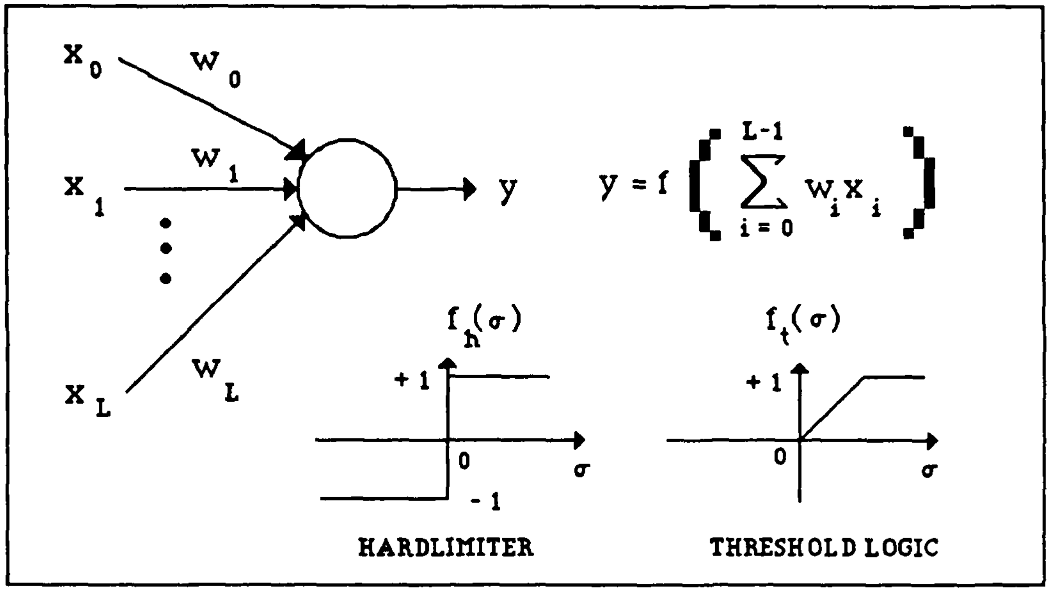


Figure 3-9. Functions used in Associative Units (Lippmann,1987:5)

The optimization criterion for the LMS algorithm is the minimization of the mean squared error between the actual and desired output. This is explained in greater detail in a later section.

In Figure 3-8, the x_n represent the elements of the **x** vector which is the feature vector to be classified. The w_n represent elements of the **w** vector which is a vector of weights used to generate decision functions. The w_n are the parameters which are updated during the training of the algorithm and ultimately are responsible for class membership decisions.. Since it has only one output node the perceptron shown in this figure can be used only for a two class problem (Tou and Gonzalez, 1974; 161). For the multiclass problem of this thesis, this structure needs to be modified.

A mulitclass perceptron algorithm is described by Tou and Gonzalez and also by Lippmann. The modification consists of adding output nodes to the structure of Figure 3-8. There is an output node for each class of feature vectors to be identified (Tou and Gonzalez,1974:181).

The scenario for the multiclass perceptron is as follows. The M pattern classes are assumed to be separable by M decision functions with the property that for an input vector x belonging to the class i (Tou and Gonzalez,1974:181)

$$d_i(x) > d_j(x) \quad \text{for all } j \neq i \tag{3-1}$$

The decision functions are defined by corresponding weights. The decision function $d_i(k)$ represents the decision function for class i at the kth iteration of the training and is given as (Tou and Gonzalez,1974:182)

$$d_i[x(k)] = w_i^T(k) \cdot x_i(k) \tag{3-2}$$

where $x(k)$ and $w_i(k)$ are the input and weight vectors. An example of this procedure is presented in the next section to illustrate the method by which the weights are updated by the process to determine the decision functions.

Example of Multiclass Perceptron Algorithm. This section demonstrates the use of the multiclass perceptron algorithm. The following example is from Tou and Gonzalez (Tou and Gonzalez, 1974:181-186).

There are M classes of patterns to be classified and are represented as $C_1, C_2, \ldots, C_M$. During the training, an input pattern $x(k)$ belonging to class C_i is presented at the kth iteration and the M decision functions are evaluated. If

$$d_i[x(k)] > d_j[x(k)] \quad j = 1, 2, \ldots, M \ ; \ j \neq i \tag{3-3}$$

then the weight vectors are not modified (Tou and Gonzalez,1974:181). This can be written as (Tou and Gonzalez,1974:181)

$$w_j(k + 1) = w_j(k) \quad j = 1, 2, \ldots, M \tag{3-4}$$

This corresponds to the situation when the optimum weights have been found and therefore the training of the classifier is completed. However, if for some decision function n (Tou and Gonzalez,1974:181)

$$d_i[\mathbf{x}(k)] \leq d_n[\mathbf{x}(k)] \tag{3-5}$$

then the weights of all the decision functions must be modified or adapted. Equations used to update the vector of weights are given as (Tou and Gonzalez, 1974:182)

$$\begin{aligned} \mathbf{w}_i(k+1) &= \mathbf{w}_i(k) + \mu\mathbf{x}(k) \\ \mathbf{w}_n(k+1) &= \mathbf{w}_n(k) - \mu\mathbf{x}(k) \\ \mathbf{w}_j(k+1) &= \mathbf{w}(k) \end{aligned} \tag{3-6}$$

where μ is a positive constant with a value between zero and one. This constant controls the speed of convergence and also affects the stability of the adaptation process (Lippmann, 1987:13). These new weights are used during the next iteration of the training process. The training is continued by applying training vectors and updating the weights until the perceptron correctly classifies all the vectors of the training set.

Relationship betweeen the Perceptron and LMS Algorithms. This section discusses the relationships between the perceptron and LMS algorithms and presents the training method used with the LMS algorithm. Equations (3-5) and (3-6) are equivalent to the hard limiter function in the associative unit shown in Figure 3-9. The weights are updated by adding $\mu \cdot \mathbf{x}(k)$ when there is a difference between the desired and actual outputs. The weights are not updated when the actual output equals the desired output. In this case, the magnitude of the difference does not affect the how the weights are updated.

As stated previously, the LMS algorithm is obtained from the perceptron algorithm by using a linear function in the associative unit shown in Figure 3-9. The weight update equation is now written as (Widrow and Stearns,1985:100)

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + 2\mu \cdot e(k) \cdot \mathbf{x}(k) \tag{3-7}$$

where

$$e(k) \quad = d(k) - \mathbf{w}_i^T(k) \cdot \mathbf{x}(k)$$

The LMS algorithm updates the weights by an amount proportional to the error bteween the desired and actual outputs.

The training of the LMS algorithm is not as straightforward as training the perceptron. Recall, the perceptron iteratively operated upon the training set until it correctly classifies each training vector. The LMS algorithm is run for M trials and for a certain number of iterations. Then, the average of $e^2(k)$ over the M trials is observed as a function of the iteration number, k. The weights are said to have converged when $e^2(k)$ does not decrease with increasing iteration number (Widrow and Stearns, 1985:105).

Although the LMS algorithm results from a small change to the perceptron algorithm, it has an important advantage over the perceptron. Lippmann states that "the perceptron convergence procedure ... may oscillate continuously when inputs are not separable and distributions overlap." (Lippmann,1987:14). The LMS algorithm will converge in this case and the result is the least mean squares solution (Lippmann,1987:14).

Application of LMS Algorithm. Figure 3-10 depicts the features extracted earlier being applied to the LMS algorithm to generate errors used to update the weight vectors. During the training portion of the classifier, the feature vectors are from signals of known modulation type. Then the classifier calculates the actual output from each weight vector. The initial weights are initially set equal to zero and are subsequently updated during the adaptation. The gain constant is determined use of a formula given by Widrow and Stearns as (Widrow and Stearns,1985:103)

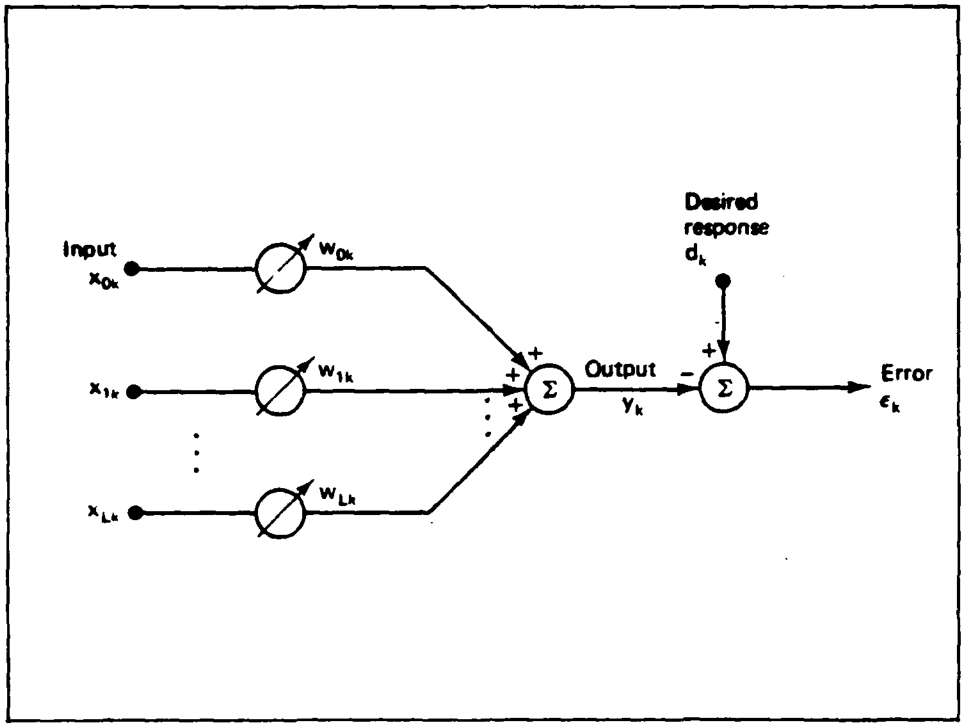


Figure 3-10. Feature Vectors Applied to Classification Algorithm (Widrow and Stearns,1985:101)

$$0 < \mu < \frac{1}{(L+1) \cdot (\text{signal power})} \tag{3-8}$$

where

$L + 1$ = number of elements in weight vector

signal power = $\mathbf{x}^T\mathbf{x}$

In practice, the value of μ is chosen to be an order of magnitude less than the upper limit given by equation (3-8) (Widrow and Stearns,1985:103).

The training consists of cyclically applying a set of known vectors from each modulation type to the classifier for a specified number of iterations. It is during this training that the elements of each weight vector converge to their final values.

After the training is completed, signals of unknown modulation types are input to the classifier which then assigns them to classes based upon evaluation of the decision functions. In this thesis the decision functions are evaluated in parallel. The class decision is made by selecting the decision function with the largest output as shown in Figure 3-11.

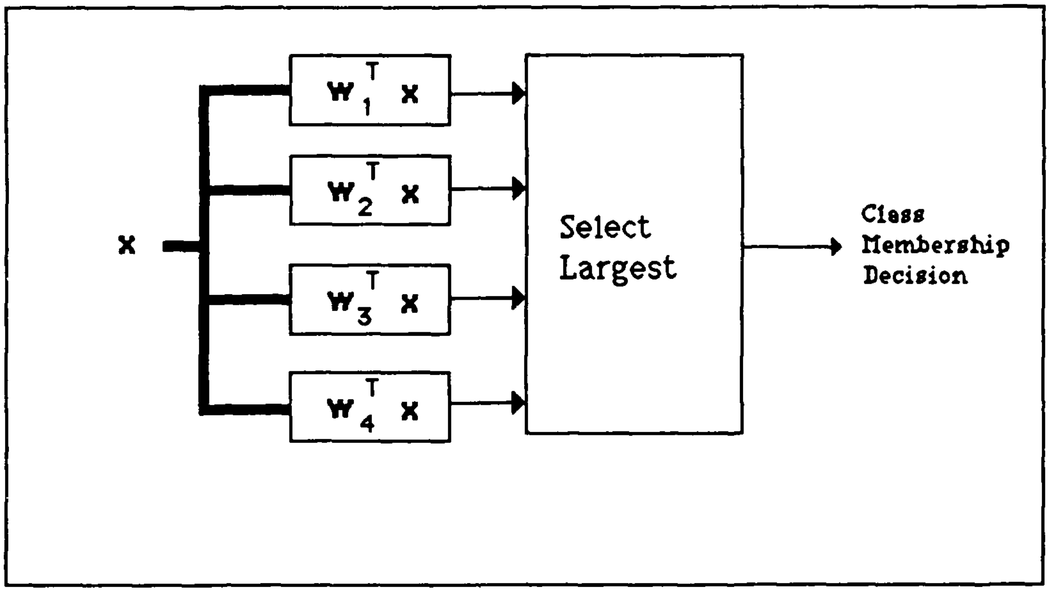


Figure 3-11. Method of Class Membership Decisions (Lippmann,1987:5)

Summary

The theory required for an understanding of the operation of this classification scheme has been presented. The classification begins with the calculation of features from the signal. The envelope statistics provide information concerning amplitude or angle modulation. The spectrum of the signal allows estimates of carrier frequency, and level of FSK. The spectra of the signal squared and quadrupled provide features for the determination of level of phase modulation.

The classification algorithm uses an adaptive procedure which first operates upon a set of feature vectors obtained from known classes to generate weight vectors. After the weight vectors have converged, the classifier is ready to operate upon unknown signals. The next chapter explains the procedure used to classify signals according to the theory presented in this chapter.

IV. PROCEDURE

Introduction

This chapter presents the procedure used during the computer simulation which performs the classification process described in the preceeding chapter. First, the sample rate and the observation interval of the computer generated signals are justified. Second, an overview of the structure used for the processing is described. This overview shows the flow of signals from the waveform stage to feature vector stage to classification in a conceptual fashion.

The steps of the feature extraction process are then described. The feature extraction process is presented here because feature vectors are needed to train the adaptive classifer. The features extracted are used as elements of the feature vectors. The construction of the feature vectors from these elements is presented. Also, the method used to train the classifier is described. Then, the classification of signals from their feature vectors is presented.

The summary reviews the major topics of the overall classification procedure. The processing software is referenced in the corresponding appendices.

Computer Generated Signals

The waveforms generated for this experiment are digitally modulated signals and noise. The parameters for each type of waveforms are presented in this section.

Generation of Digitally Modulated Signals. The signals segments used in this experiment consist of 8192 samples with an intersample period of 1 microsecond. The symbol rate for all signals is 2500 symbols per second. This results in 8.192 milliseconds of data which corresponds to 20.48 symbols per observation interval. The need for baseband sampling as opposed to bandpass sampling is discussed in a later section.

The center frequency for BASK, BPSK, and QPSK signals is 100 kHz. This was selected for convenience and is not a typical intermediate frequency of a receiver. However, this does not affect the performance of the classification procedure. The frequencies of the BFSK signal are 80 and 100 kHz and were again chosen for convenience. The programs which generate BASK, BPSK, QPSK, and BFSK are named OOKGEN, BPSK,QPSKGEN,and FSKGEN. They are listed in Appendix B.

Generation of Noise. The noise used in this experiment was additive white gaussian noise which was generated by summing 50 random vectors whose elements were uniformly distributed over -0.5 to 0.5. The resultant vector has 8192 elements with a gaussian distribution of zero mean and unity variance. The unity variance was achieved by scaling the elements. This random vector is then used as a noise waveform which is added to the signals generated above. The desired signal to noise ratios are obtained by scaling the amplitude of the carrier waveform to the desired values. The program which generates noise is named GAUSS and is listed in Appendix B.

Structure of the Procesor

The signal flow through the feature extraction and classification steps are the same for all types of signals. The structure of the processor is shown in Figures 4-1 and 4-2. The nine elements of the feature vectors are obtained from the signal's envelope, the spectrum of the signal, the spectrum of the signal raised to the second power, and the spectrum of the signal raised to the fourth power. Assuming the classifier has been trained and has valid weight vectors, the feature vectors are then used as inputs to the classifier which performs the classification as explained in the previous chapter.

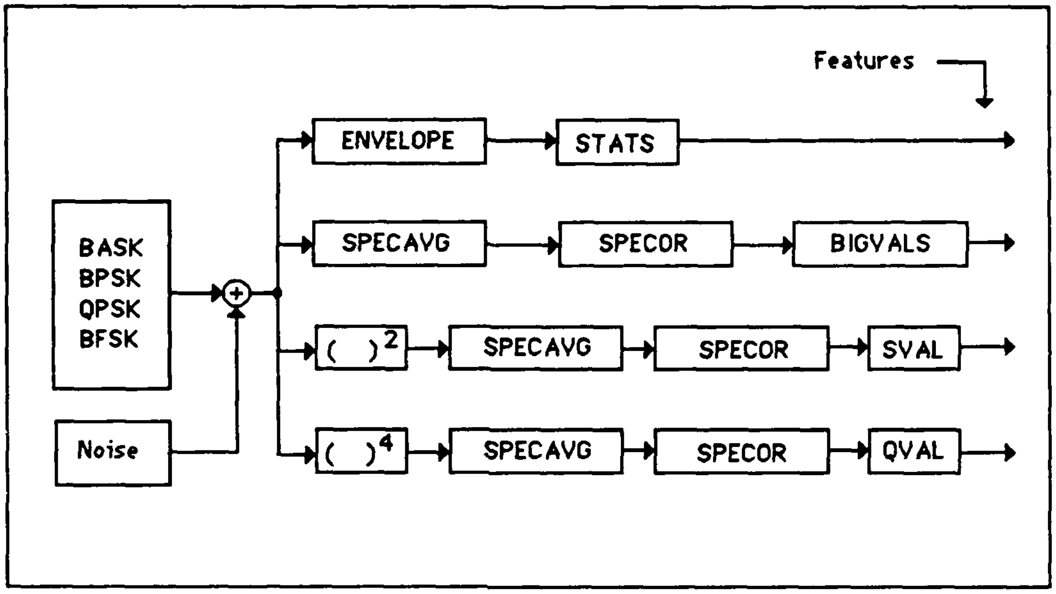


Figure 4-1. Structure of the Classification Procedure

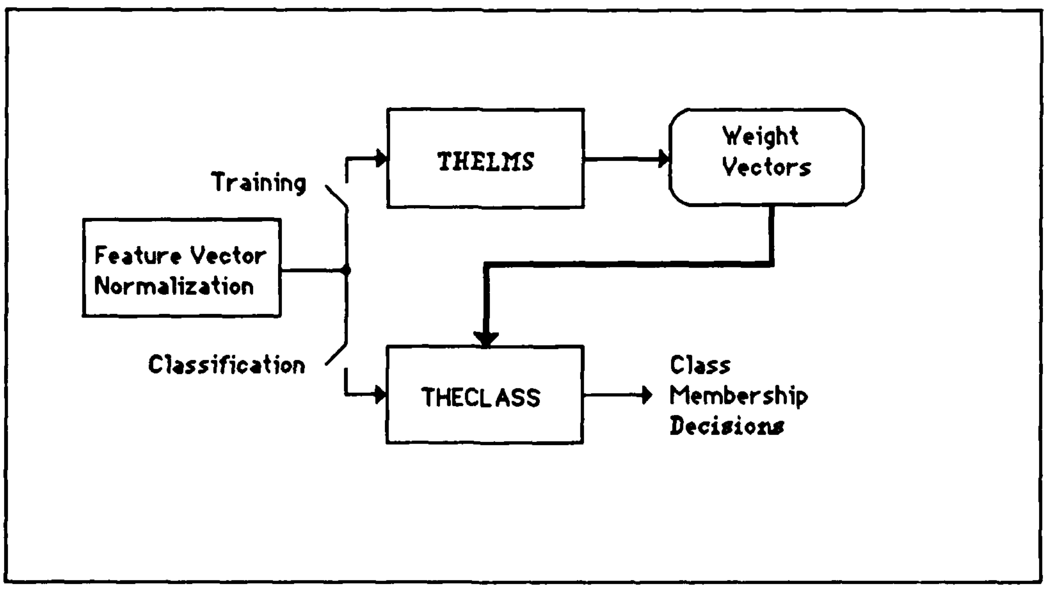


Figure 4-2. Training and Classification

Feature Extraction

The features used in this classification procedure are calculated from four fundamental operations upon the unknown signal. One of the operations is the calculation of the mean and variance of the envelope of the waveform. Another operation is the searching of the spectrum of the signal for the two largest peaks. The third operation searches the spectrum of the signal squared for peaks near twice the intermediate frequency. The fourth operation searches the spectrum of the signal raised to the fourth power for peaks near four times the intermediate frequency. Each of these operations are described below.

Features from the Envelope. The first processing function calculates the envelope of the waveform. This is accomplished by the program ENVELOPE, which is listed in Appendix B. The mean and variance of the envelope are then calculated by the program STATS which is listed in Appendix B. The mean and variance of the envelope are the first two elements of the feature vectors.

Features from the Spectrum of the Signal. The next step in the processing is to calculate the spectra of the signal. The spectra calculated here are the result of averaging two 4096 point spectra. The 1 MHz sample rateresults in frequency bins of 244.140625 hertz. Rectangular windowing is used on the data and it is then passed to the FFT subroutine in the program SPECAVG. Rectangular windowing was chosen over any other windowing since it provides the least amount of spreading of spectral energy (Rabiner and Gold,1975:95). SPECAVG is listed in Appendix B.

The resultant spectra are correlated with a $\mathrm{sinc}^2(x)$ function which has a null to null bandwidth of 5 kHz. This bandwidth corresponds to the theoretical bandwidth of all the signals considered. Before the correlation is performed in the program SPECOR, both spectra (the magnitude spectrum of the signal and the $\mathrm{sinc}^2(x)$ spectrum) are normalized to unity energy. This normalization is necessary in order for all correlation values to range from zero to unity. SPECOR is listed in Appendix B.

The features obtained from this stage are selected by the program BIGVALS. This searches the result of the preceeding correlation for the two largest values. These values and their spectral locations provide the next four elements of the feature vectors. The search ignores points within ten points of the largest value in order for the search to ignore large values from the same spectral lobe. BIGVALS is listed in Appendix B.

Feature from the Spectrum of the Signal Squared. The next step is to calculate the magnitude spectrum of the signal raised to the second power. This is the step intended to provide information related to BPSK signals. The resultant spectrum is correlated with a $\mathrm{sinc}^2(x)$ function of 1 kHz null to null bandwidth. Although the search is for narrowband energy, consistent detection of energy near twice the intermediate frequency was obtained without using a smaller bandwidth $\mathrm{sinc}^2(x)$ function. Recall, in the previous chapter this value was specified to be determined empirically. Satisfactory results were obtained with this bandwidth of the reference function.

The program which performs the correlations sets the dc portion of the spectrum to zero prior calculating its energy which is also prior to the correlation. This is to eliminate the response near zero hertz due to squaring and quadrupling the signal.

The result of the correlation is searched for a peak amplitude. However, in this case, only points within a range of 100 kHz of twice the carrier frequency are considered in the search. The feature obtained from this procedure is the amplitude of the peak within the search range. The spectral location was the same for all signals of the training set; therefore, this would not provide information useful for the separation of classes. The program which performs this search is named SVAL and is listed in Appendix B.

Features from the Spectrum of the Signal Quadrupled. The next two features are obtained from the spectrum of the signal raised to the fourth power. This step of the processing provides information related to QPSK signals. The spectrum of the signal which has been raised to the fourth power is searched for the largest value near four times the intermediate frequency. The value of the correlation peak and its spectral location are used as the eighth and ninth elements of the feature vectors. The program which does this search is named QVAL and is listed in Appendix B.

The need to perform the sampling as baseband and not bandpass is explained by noting that the features extracted in the above steps were dependent upon the intermediate frequency and its second and fourth multiples. Had the signals been bandpass sampled, the information related to the intermediate frequency would have been lost.

Construction of the Feature Vectors. The feature vectors consist of elements whose values may range from on the order of unity to the order of thousands. For example, all the peak correlation values will be less than one, while the spectral location of the correlation peak of the signal raised to the fourth power is above 1600. This number is the FFT bin number, not a frequency value. In order to prevent this one element from dominating the adaptation and classification procedure, all elements are scaled to range from zero to one.

The method used in this experiment to normalize the feature vectors is now explained. The normalization is performed over the sets of signals grouped according to SNR. In practice, the normalization could be performed over the signals collected during one event if off line classification were feasible. Near realtime classification would require that the elements be scaled to values between zero and one before constructing the feature vectors.

The normalization operates upon the same element of each feature vector at a time. The first element of each vector is searched for the highest and lowest values. For example, assume the highest value is a and the lowest value is b. The range is found by subtraction to equal a - b. Then b is subtracted from each element. The result of this subtraction is then divided by the range. This normalization provides elements between the values of zero and one for each element of all the feature vectors.

Training the Classifier

The training of the classifier is performed by using feature vectors which are calculated from signals whose class is known. The signals used for training in this experiment had 20 dB signal to noise ratios. One feature vector is calculated from each class. These feature vectors are then cyclically applied to the LMS algorithm for a fixed number of iterations.

The output of the algorithm is a weight vector for each class of signal considered. During training, the desired response is a function of the input feature vector and the weight vectors used. This is illustrated in Figure 4-3. The program which uses this algorithm is named THELMS and is listed in Appendix B.

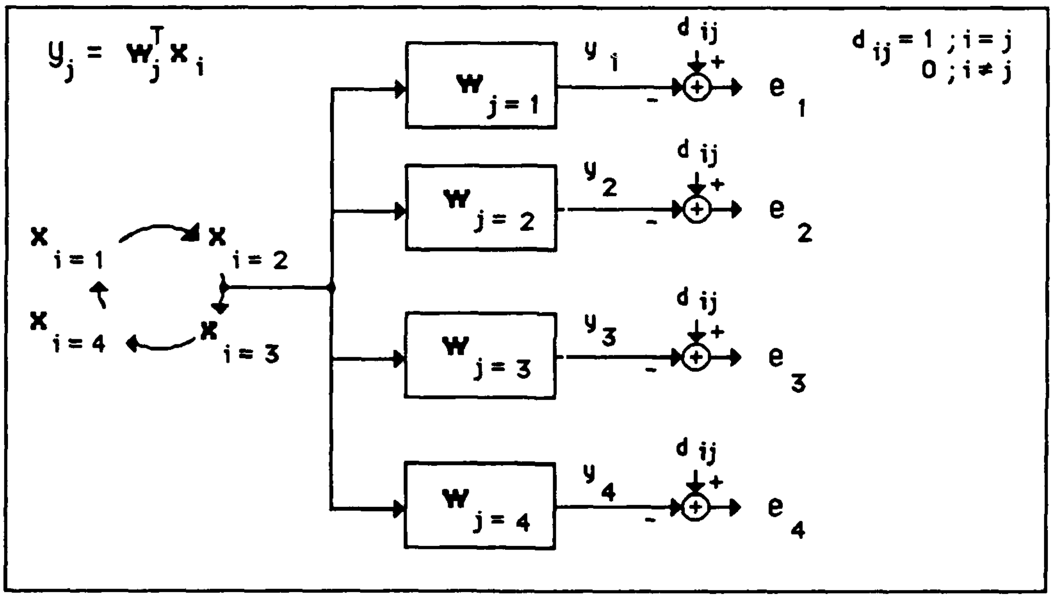


Figure 4-3. Training Using the LMS Algorithm

The feature vectors which have been calculated from known classes are cyclically applied to the algorithm. The weight vectors for each class of signals are updated during each iteration. The procedure is shown in Figure 4-3. The equation used to update the weights is given in equation (4-1) as (Widrow and Stearns,1985:103)

$$\mathbf{w}_i(k + 1) = \mathbf{w}_i(k) + 2\mu[d_{ij} - y(k)] \cdot \mathbf{x}_j(k) \tag{4-1}$$

where

i	=	class indicator for weight vectors
j	=	class indicator for feature vectors
$\mathbf{w}_i(k+1)$	=	weight vector for class i at next iteration
$\mathbf{w}_i(k)$	=	weight vector for class i at present iteration
μ	=	gain constant
$d_{ij}(k)$	=	$1;\ i = j$
	=	$0;\ i \neq j$
$y(k)$	=	$\mathbf{w}_i^T(k) \cdot \mathbf{x}_j(k)$
$\mathbf{x}_j(k)$	=	feature vector from class j

This algorithm is applied to the feature vectors generated from 20 dB SNR signals. The convergence of the weight vectors are confirmed by running several trials with different numbers of iterations and different values of the gain constant. The weight vectors used in this experiment are calculated from 100000 iterations of the LMS algorithm with a gain constant of 0.001185. The convention for specifying class membership is that BASK, BPSK, QPSK, and BFSK belong to class 1, class 2, class 3, and class 4.

Classification of Unknown Signals

The weight vectors calculated in the previous section are used in the classification of unknown signals as shown in Figure 4-4. A program named THECLASS performs the "select largest" function of the figure. THECLASS is listed in Appendix B.

The unknown signal is processed to generate a feature vector, shown in the figure as $\mathbf{x}_i$. This feature vector is then multiplied with the four weight vectors. These are the weight vectors calculated by the LMS algorithm during the training. The equations used in this classification process may be written as

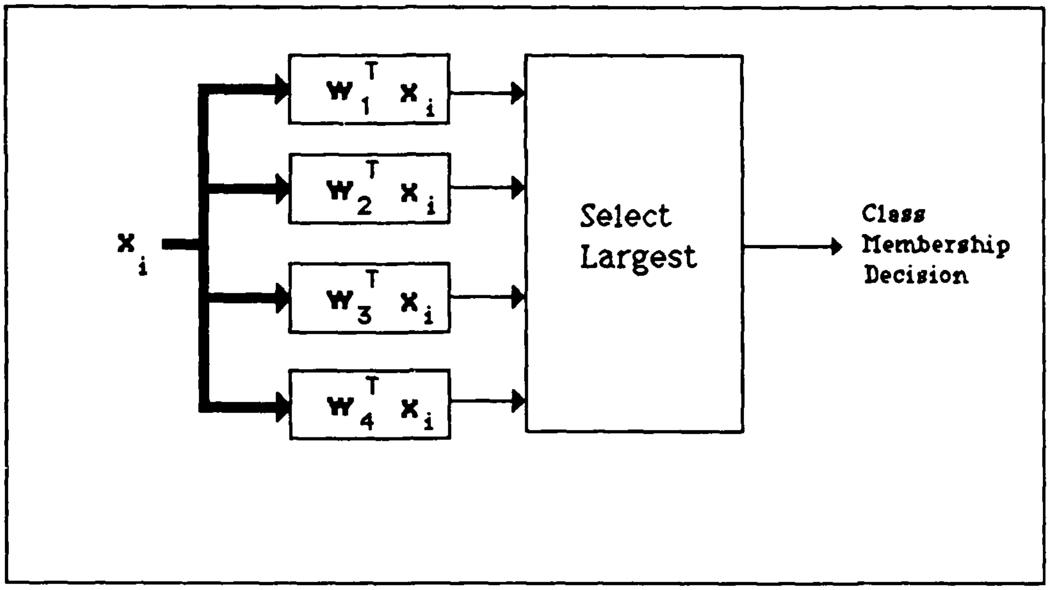


Figure 4-4. Classification Using Weight Vectors

$$d = w_i^T \cdot x_j \ ; \quad i = 1,2,3,4 \quad \text{and} \quad j = 1,2,3,4 \qquad (4\text{-}2)$$

Class memebership is determined by selecting the class which corresponds to the weighting function which produces the largest output.

Summary

In this chapter, the classification process has been presented. Details of the feature extraction, training, and classification portions have been given along with the program names which perform the calculations. The results of applying this procedure to the four classes of signals is presented in the next chapter.

V. RESULTS

Introduction

This chapter presents the results of the experiment performed to classify signals according to modulation type. The procedure used has been described in the previous chapter. In this chapter, the parameters used for the generation of the modulated signals are given. Then, the feature vectors generated from these signals are presented. The next section shows the results of using the weights obtained by the LMS algorithm to perform signal classification. This chapter concludes with a summary of the results of this classifier on the signal set.

Signal Generation

There are five basic sets of signals used in this experiment. The first set of signals is used for training the classifier and the remaining four sets are used to test the performance of the classifier on unknown signals. The convention for specifying class membership is that BASK, BPSK, QPSK, and BFSK belong to class 1, class 2, class 3, and class 4.

Generation of Training Signals. The signals used to trained the classifier consist of one sample from each of the classes considered in this experiment. The SNR for this set is 20 dB. This value is obtained by scaling the amplitude of the carrier. The power of the signal classes considered here is calculated in equation (5-1) as (Gagliardi,1978: 19)

$$P = A^2/2 \tag{5-1}$$

where

P = peak signal power

A = amplitude of sinusoidal carrier

Since the noise has unity variance and is zero mean, its power is equal to one. Therefore, the SNR in dB is computed as (Gagliardi,1978:20)

$$SNR = 10 \log(A^2/2) \tag{5-2}$$

Alternately, the amplitude of the carrier can be written as a function of signal to noise ratio by rearanging equation (5-2). Doing so, we obtain

$$A = 2^{1/2} \cdot 10^{SNR/20} \tag{5-3}$$

For example, to obtain a SNR of 20 dB, the amplitude of the carrier is found to be approximately 14.14 volts. The resultant noisy waveform used in the classification procedure is obtained by adding a file of noise points to the file of modulated data points. The same noise file is used to corrupt each of the waveforms in the training set.

The modulating data is 21 consecutive bits chosen from a pseudonoise sequence. The data for BASK, BPSK, and BFSK consists of the same 21 bits. Since QPSK bauds convey two bits per symbol as opposed to the binary modulation schemes, more data bits are required to obtain the same observation interval as the other signal classes. Therefore, 42 bits are used, with the first 21 bits being the same as the binary modulation schemes. The next 21 bits of the pseudonoise sequence are used to obtain the second half of the QPSK data. Segments of the waveforms at 20 dB SNR from each class of signals are shown in Figures 5-1 through 5-4. Feature vectors are then calculated from these four samples of signals as described in the previous chapter.

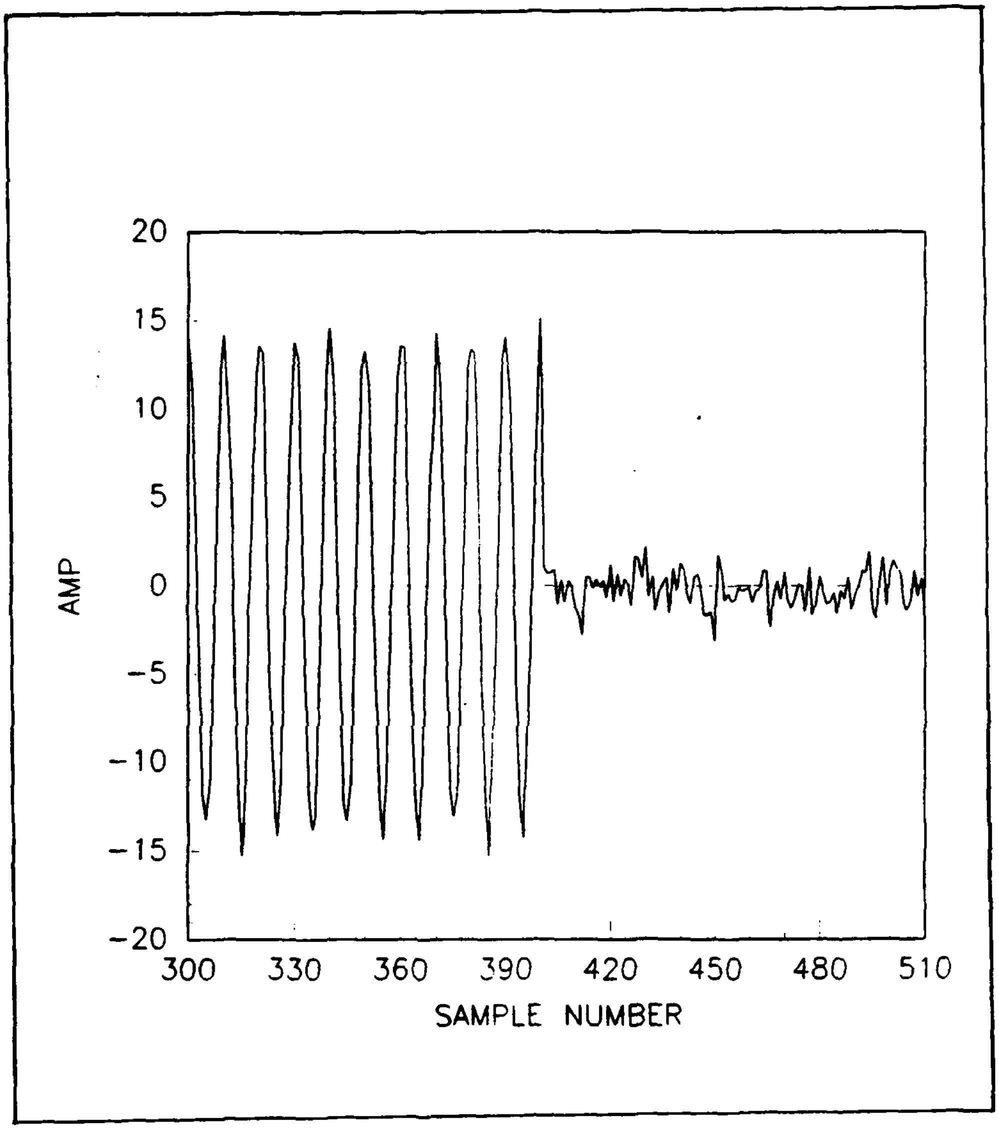


Figure 5-1. Sample of 20 dB BASK

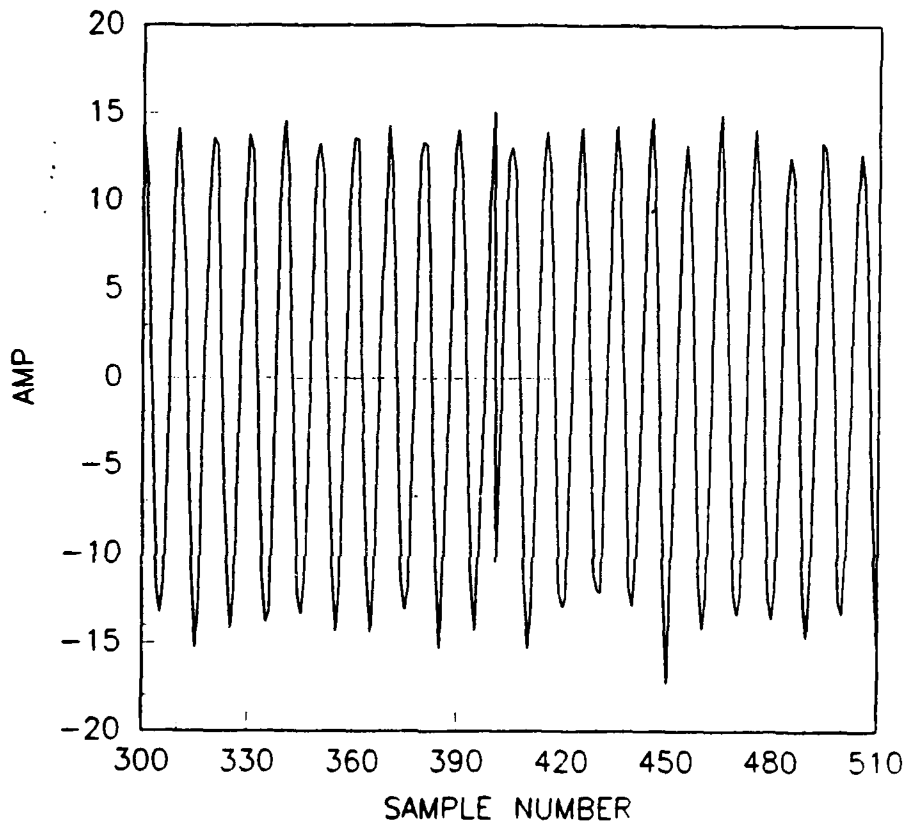


Figure 5-2. Sample of 20 dB BPSK

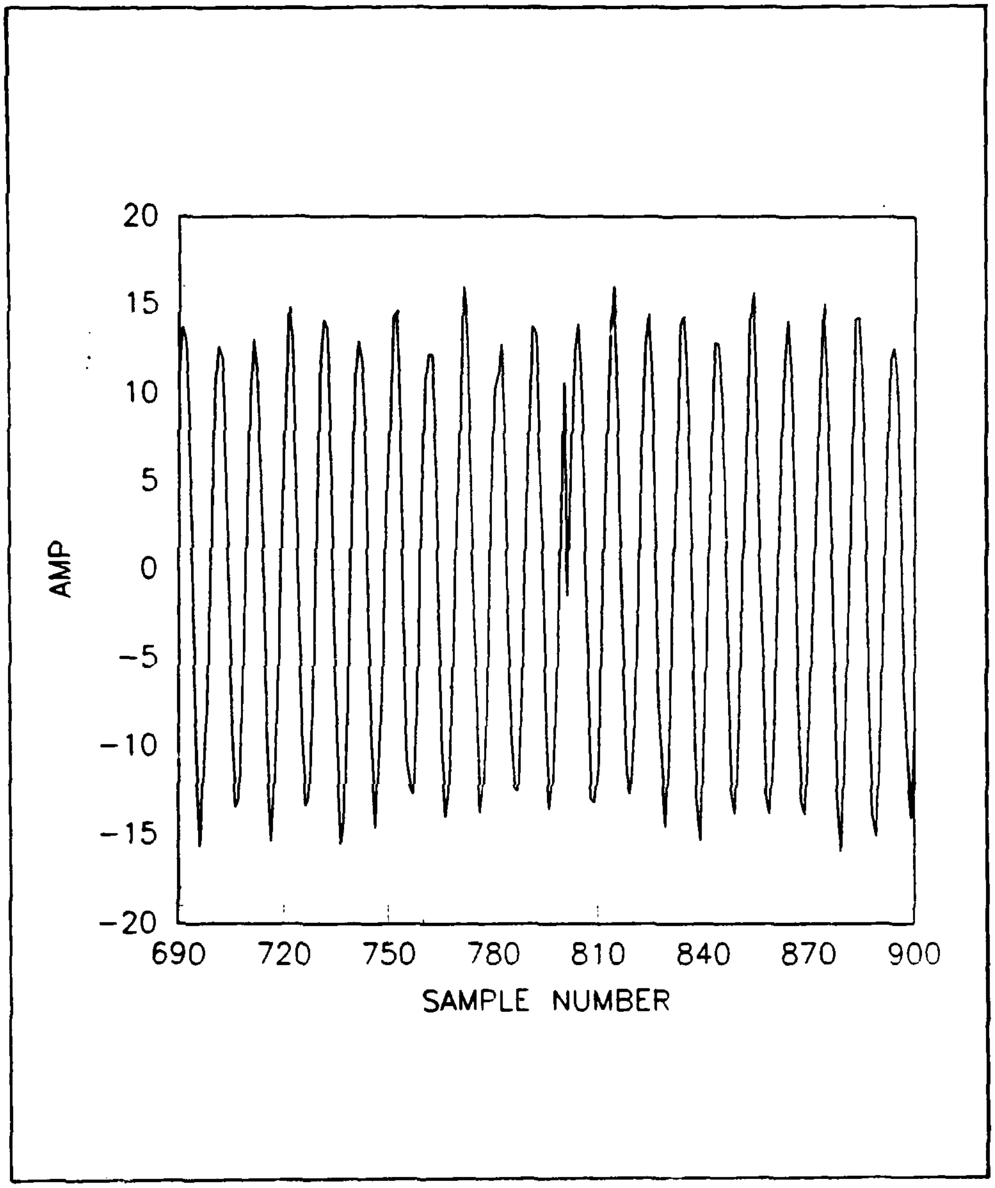


Figure 5-3. Sample of 20 dB QPSK

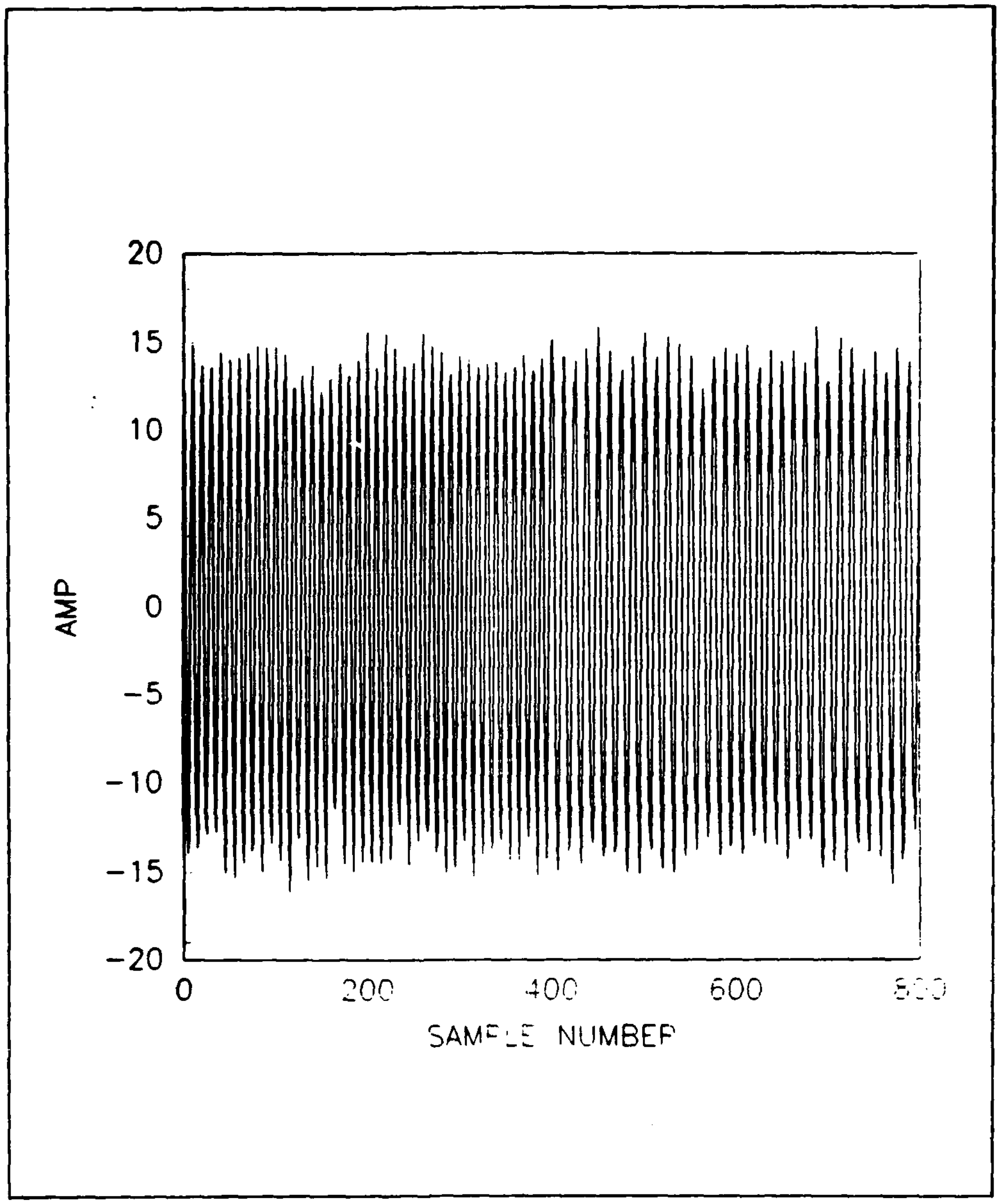


Figure 5-4. Sample of 20 dB BFSK

Generation of the Unknown Signals.

The unknown signals are generated in a similiar fashion as the signals used in the calculation of the feature vectors. The differences are that the underlying data is different from the training set and a different noise file is used to corrupt the signals. Another set of signals from each class is generated at 20 dB SNR. The underlying data is different than the training set and the noise comes from a different nose file of unity variance.

The next set of signals is generated in the same manner as above but the amplitude is scaled to acheive a 15 dB SNR, the data bits are different than from the first two sets of signals and a new noise file is used. The fourth set of signals is generated at a 10 dB SNR with new data bits and a new noise file. The fifth set of signals is generated at a 5 dB SNR with new data bits and a new noise file.

Figures 5-5 through 5-7 show BASK and BPSK waveforms at the 15, 10, and 5 dB SNRs considered in this experiment. These classes of signals are chosen in order to illustrate the effect of noise on the waveforms of amplitude and angle modulated signals.

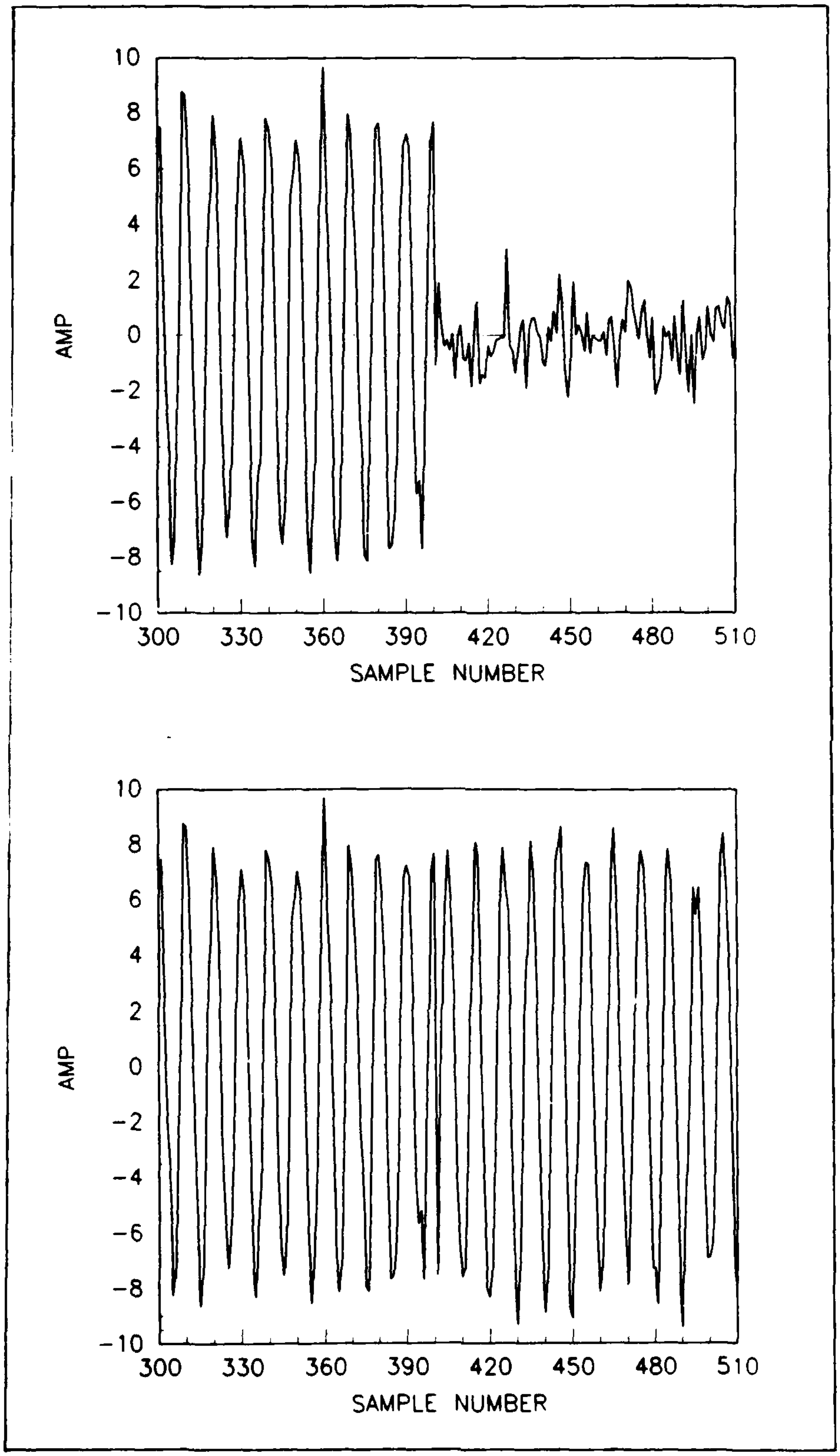


Figure 5-5. Samples of BASK and BPSK at 15 dB

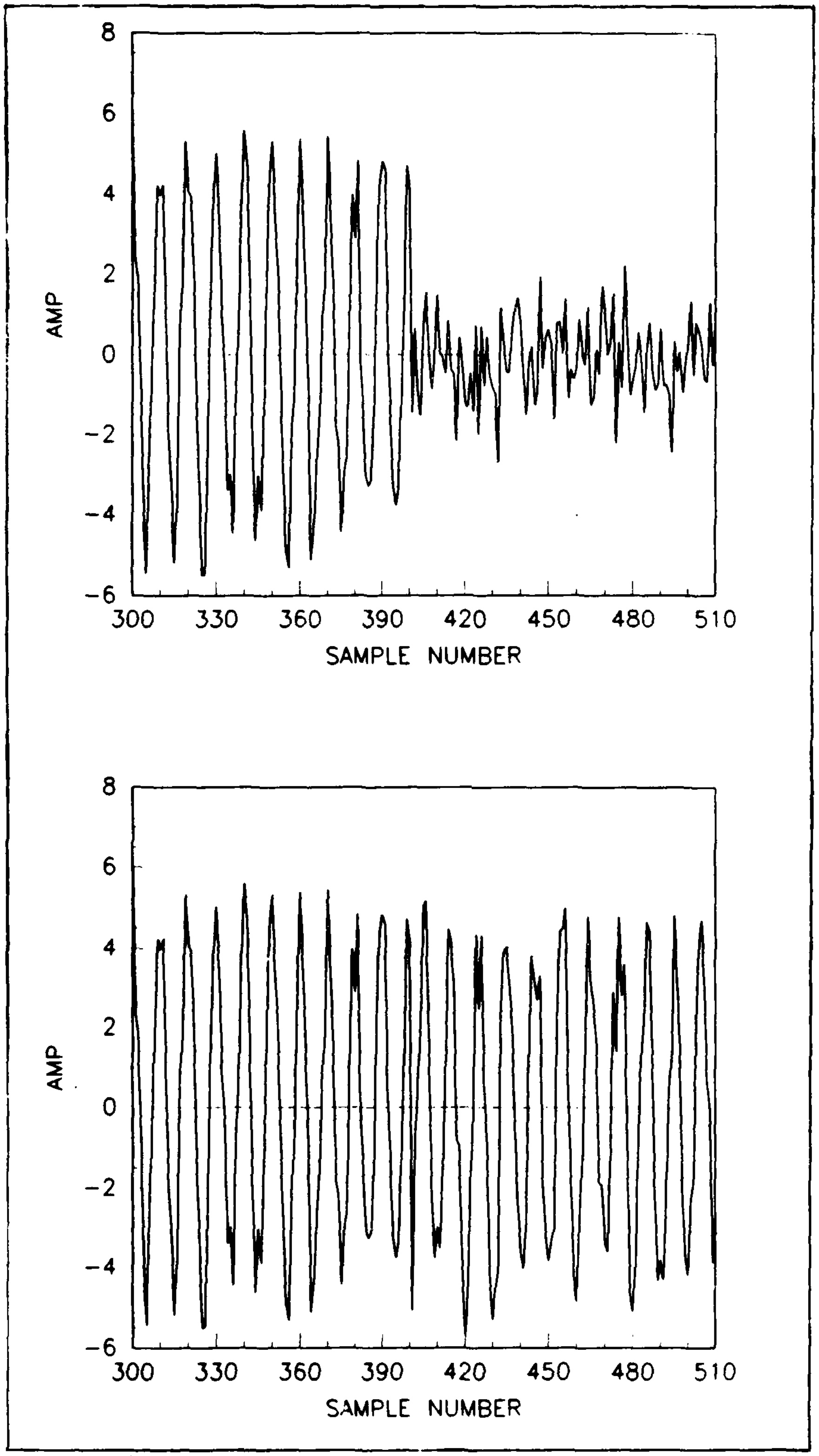


Figure 5-6. Samples of 10 dB BASK and BPSK

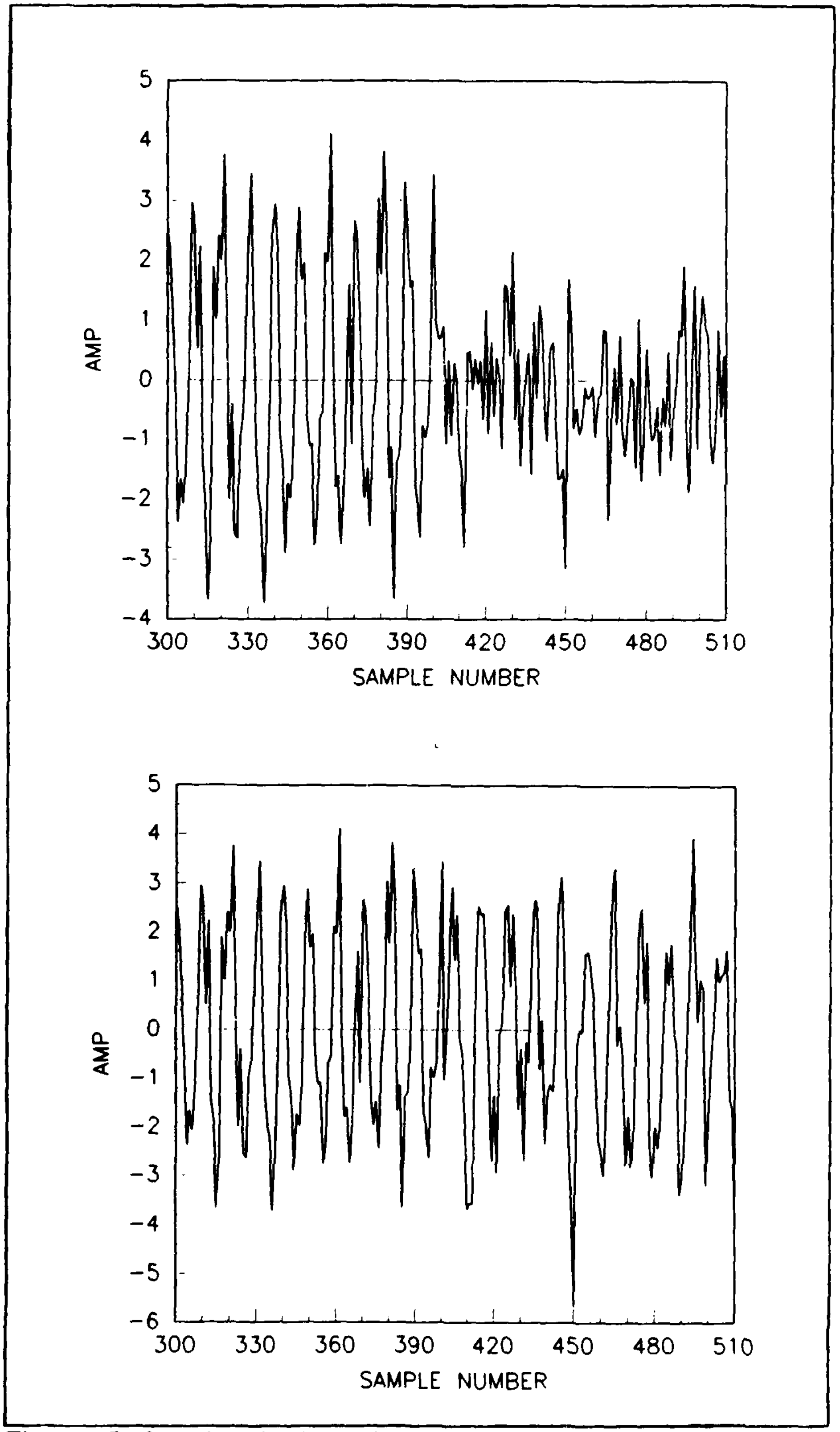


Figure 5-7. Samples of 5 dB BASK and BPSK

Calculation of Feature Vectors

The signals are operated upon by the feature extraction and feature vector normalization processes described in the previous chapter. Feature vectors calculated from the four different sets of signals are presented in Table 5-1. When these feature vectors

Table 5-1. Feature Vectors from 20 dB SNR Signals

Description	BASK	BPSK	QPSK	BFSK
mean	0.00000	0.99721	1.00000	0.99400
variance	1.00000	0.00126	0.00429	0.00000
maximum 1	1.00000	0.04105	0.66320	0.00000
location 1	1.00000	0.97559	0.98782	0.00000
maximum 2	0.00000	0.19540	0.06738	1.00000
location 2	0.00000	0.88890	0.94447	1.00000
squared	0.40009	1.00000	0.00000	0.31237
quadrupled	0.00700	0.15270	1.00000	0.00000
augment	1.00000	1.00000	1.00000	1.00000

are used to train the classifier, the weight vectors of Table 5-2 are generated.

The description columns of the tables refer to the feature extraction portion of the experiment. The first and second elements in each feature vector are related to the mean and variance of the envelope. BASK has the smallest mean and largest variance. The third and fourth elements are the result of the correlation of the spectrum of the signal with the $\mathrm{sinc}^2(x)$ reference function. The elements correspond to the correlation value and offset to this value, respectively. The fifth and sixth elements of the vectors are similar to the third and fourth, except they are related to the second largest correlation value and its offset. The seventh element is derived from the correlation of the spectrum of the signal squared with the $\mathrm{sinc}^2(x)$ reference function. It corresponds to the largest correlation value found near twice the intermediate frequency. The eighth element is similiar to the seventh except it is the result of searching the correlation of the spectrum of the signal raised to the fourth power with the

Table 5-2. Weight Vectors

Description	BASK	BPSK	QPSK	BFSK
mean	-0.18255	0.12045	0.13634	0.09367
variance	0.40205	-0.19119	-0.14394	0.15369
maximum 1	0.29829	-0.36294	0.21310	0.06203
location 1	0.07248	0.35930	0.10177	-0.39277
maximum 2	0.10760	-0.29654	-0.13564	0.56082
location 2	-0.15284	0.03968	0.13853	0.15205
squared	0.01894	0.66547	-0.40747	-0.18181
quadrupled	-0.17228	-0.18061	0.51185	-0.17937
augment	0.22078	-0.07073	-0.01114	0.25080

$\mathrm{sinc}^2(x)$ function. The value is related to the largest correlation peak found near four times the intermediate frequency. The ninth element is called the augmentation of the feature vectors. This constant value allows the LMS algorithm to account biases in the feature vectors (Widrow and Stearns,1985:17).

Classification Results.

The results when the feature vectors are multiplied with the weight vectors are shown in Table 5-3. For each class of signal, the largest result of the multiplication occurs when the weight vector is matched to the class of signal.

Table 5-3 Classification Results

	$w_1^T x$	$w_2^T x$	$w_3^T x$	$w_4^T x$
20 dB				
BASK	0.8300	-0.0592	0.2167	0.1728
BPSK	0.0875	0.9702	-0.1045	-0.0272
QPSK	0.1726	-0.0862	0.8996	-0.1609
BFSK	0.3508	-0.4222	0.1772	1.0070
15 dB				
BASK	0.7107	-0.0734	0.2521	0.1771
BPSK	0.0379	0.9505	0.0014	-0.0170
QPSK	-0.0582	0.2265	0.7370	-.2281
BFSK	0.4026	-.1883	0.0012	0.9091
10 dB				
BASK	0.8544	-0.0302	0.1699	0.1641
BPSK	0.1887	0.7744	0.1770	0.0132
QPSK	-0.1027	-0.2071	0.6016	0.3066
BFSK	0.3266	0.2093	0.1643	0.5617
5 dB				
BASK	0.9609	-0.1639	0.0836	0.0637
BPSK	0.0375	1.0678	-0.566	-0.1530
QPSK	0.0375	0.0.0125	0.9026	-0.0579
BFSK	0.1618	-0.6281	0.6975	0.9498

<u>Summary</u>

This chapter has presented the results of a classification experiment which was designed to spearate four classes of signals. The modulation types considered were BASK, BPSK, QPSK, and BFSK. A set of feature vectors were calculated for 20 dB SNR signals from each class. These feature vectors were used for training the classifier. The LMS algorithm was used to calculate weight vectors used to classify 16 unknown signals.

The 16 signals consisted of one sample of each signal class at 20, 15, 10, and 5 dB SNRs. Different data symbols and noise files were used in the generation of the signals at different SNRs. The classification procedure correctly identified all 16 signals.

The final chapter of this thesis presents some conclusions about this classification procedure and some recommendations for further study.

VI. Conclusions

Introduction

This chapter presents some of the conclusions applicable to the classification procedure explained in the preceeding chapters. Then, some recommendations for further study are discussed.

Conclusions

The technique presented here uses features which are calculated using conventional signal processing methods and shows favorable classification properties for the following classes of signals: BASK, BPSK, QPSK, and BFSK. The amount of preprocessing required for feature extraction is comparable to the preprocessing required by the classifiers due to Liedtke and Jondral (Liedtke,1984; Jondral,1985). The number of sample points and the observation intervals are comparable between all techniques presented here.

The most interesting conclusion is that a new feature for the identification of the number of phase states of a phase shift keyed signal has been shown to provide adequate information to identify BPSK and QPSK at SNRs down to 5 dB. However, the conclusiveness of these results are limited due to the small number of signals used.

Recommendations for Further Study

As recommendations for further study, several options should be considered. The purposes of the recommendations are to provide additional information about the performance of the classification procedure. These recommendations are presented below.

Larger Set of Signals. The first recommendation is that the classifier be tested with hundreds of signals from each class. Different noise and data files should be used during this testing. Then, the results of the classification procedure would be more conclusive. The researchers of the literature review use this order of magnitude of signals during performance testing of their classifiers.

Estimate Bandwidth of Signal. The second recommendation is that methods be explored to estimate the bandwidth, and thereby the symbol rate, of the unknown signal. One possible source of this information is the result of the correlation of the spectrum of the signal with the $\text{sinc}^2(x)$ function. The bandwidth of the major peak is related to the width of the central lobe and to the width of the $\text{sinc}^2(x)$ function. Since the width of the $\text{sinc}^2(x)$ function is known, the bandwidth of the unknown signal could be calculated.

Simulated Environment. A more realistic signal environment would be another factor to consider in order to fully test this technique. In addition to AWGN, single and multiple interferers should be considered and their effects upon classification performance measured. Types of inteference should include continous wave signals, nearby analog modulated signals, and nearby digitally modulated signals. Performance of the classification procedure as a function of the strength and frequency offset of the interferers could then be measured. In addition, the effect of shaping the pulses used during modulation should be investigated.

Additional Modulation Types. This classification technique was tested on four classes of signals. By straightforward extension of the ideas presented here, this method should be able to classify 8-PSK and QFSK. In addition, this method could be tested for its ability to classify minimum shift keyed (MSK) signals. MSK signaling is a form of BFSK where the frequency separation is the smallest amount possible to obtain orthogonal signaling waveforms (Cooper and McGillem,1986:187) Alternatively, MSK may be viewed as a special case of QPSK (Stremler,1982:599). The phase of the MSK waveform changes by $\pi/2$ during each symbol interval (Stremler,1982:600). Therefore, MSK may be classified as BFSK or QPSK. The goal of this recommendation is to investigate a method which classifies MSK not as BFSK or QPSK but as MSK. There is a possibility that the method presented in this thesis may be able to perform this classification.

Bibliography

Ahmed, Nasir and T. Natarajan. Discrete-Time Signals and Systems. Reston, Virginia: Reston Publishing Company, Inc., 1983.

Chan, Y.T. and others. "Identification of the Modulation Type of a Signal," Proceedings of the IEEE 1985 International Communications Conference. 22.5.1- 22.5.4. New York: IEEE Press, 1985.

Cooper, George R. and Clare D. McGillem. Modern Communications and SpreadSpectrum. New York: McGraw-Hill Book Company, 1986.

Couch, Leon W. Digital and Analog Communication Systems. New York: Macmillan Publishing Co., Inc., 1983.

Gagliardi, Robert M. Introduction to Communications Engineering. New York: John Wiley & Sons, 1978.

Gardner, William A. Introduction to Random Processes With Applications to Signals and Systems. New York: MacMillan Publishing Company, 1986.

Hodgman, Charles D. CRC Standard Mathematical Tables (Twelfth Edition). Cleveland, OH: Chemical Rubber Publishing Company, 1959.

Jondral, Freidrich. "Automatic Classification of High Frequency Signals," Signal Processing, 9: 177-190 (October 1985).

Liedtke, F. F. "Computer Simulation of an Automatic Classification Procedure for Digitally Modulated Communication Signals with Unknown Parameters," Signal Processing, 6: 311-323 (August 1984).

Lippmann, Richard P. "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine: 4-22 (April 1987).

Proakis, John G. Digital Communications. New York: McGraw-Hill Book Company, 1983.

Rabiner, Lawrence R. and Bernard Gold. Theory and Application of Digital Signal Processing. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1975.

Schwartz, Mischa. Information Transmission, Modulation, and Noise. New York: McGraw-Hill Book Company, 1980.

Stremler, Ferrel G. Introduction to Communication Systems (Second Edition). Reading, Massachusetts: Addison-Wesley Publishing Company, 1982.

Tou, Julius T. and Rafael C. Gonzalez. Pattern Recognition Principles. Reading, Massachusetts: Addison-Wesley Publishing Company, 1979.

Widrow, Bernard and Samuel D. Stearns. Adaptive Signal Processing. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1985.

Ziemer, R.E. and W.H. Tranter. Principles of Communications. Boston: Houghton Mifflin Company, 1976.

Appendix A: Signal to Noise Ratio Analysis of Fourth Law Device

Introduction

In the "Features from Spectra of Signal Raised to Powers" section of the Theory chapter, a signal is applied to a fourth law device. A signal to noise ratio analysis of such a device is presented here.

Calculation of Signal to Noise Ratio at the Input. The input signal to the fourth law device is represented as

$$x(t) = s(t) + n(t) \tag{A-1}$$

where

$s(t) =$ waveform of modulated signal

$n(t) =$ narrowband gaussian noise with zero mean and variance v^2

The ouput signal is represented as $y(t)$. These relationships are shown in Figure A-1. First, the SNR at the input to the device is calculated. The signal power is given as (Gagliardi, 1978:19)

$$P = A^2/2 \tag{A-2}$$

Since the noise is zero mean, its power is equal to its variance (Ziemer and Tranter,1976:224). This is written as

$$N = v^2 \tag{A-3}$$

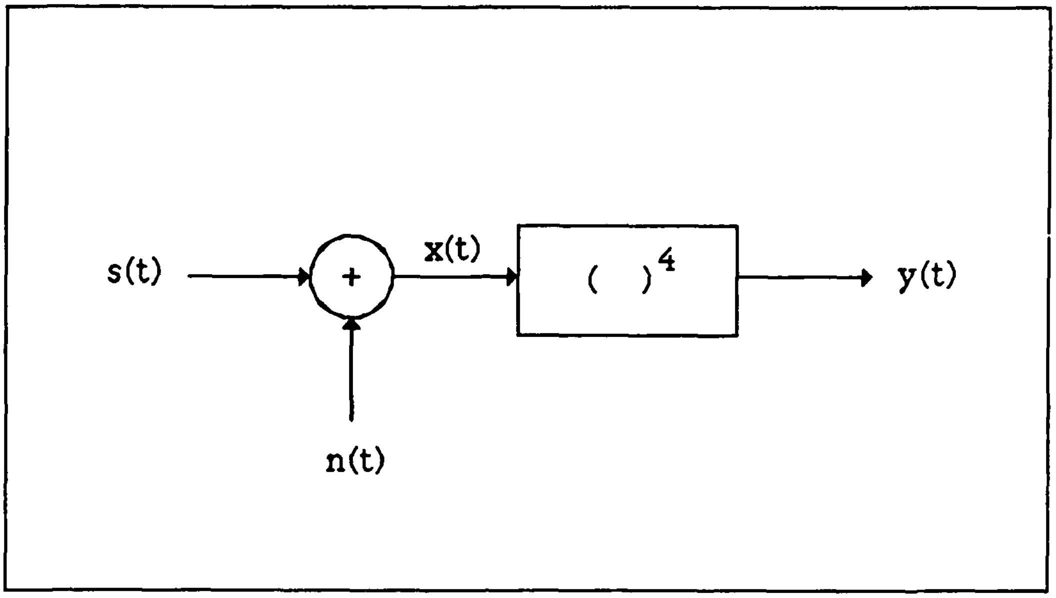


Figure A-1. Signal and Noise Applied to Fourth Law Device

The signal to noise ratio at the input is now formed by dividing equation (A-2) by equation (A-3). This results in

$$SNR_{in} = A^2/2v^2 \tag{A-4}$$

The output signal to noise ratio is calculated in the next section.

Calculation of the Signal to Noise Ratio at the Output. In order to calculate the output signal to noise ratio, the power in the noise at the output of the fourth law device must be calculated. Since the problem leading to this appendix concerns the noise power at some frequecy other than dc, the variance will be calculated since this is equal to the ac power. This will be done assuming a gaussian zero mean noise process with variance of v^2 at the input to the device. Let Z represent the random variable at the output of the device. Then, Z may be written as

$$Z = n^4(t) \qquad \text{(A-5)}$$

The variance of Z is now calculated. Using the fundamental theorem of expectation (Gardner,1986:29)

$$E\{g(x)\} = \int_{-\infty}^{\infty} g(x) \cdot f(x)\ dx \qquad \text{(A-6)}$$

In this case g(x) represents the fourth law device and f(x) is the zero mean gaussian probability density function with variance v^2. In order to find the variance of Z, the formula $Var(Z) = E\{ Z^2 \} - E^2\{ Z \}$ will be used (Ziemer and Tranter,1976:224). First, E{ Z } will be calculated. Substituting these relationships into (A-6) gives

$$E\{ Z \} = [1/(2\pi v^2)]^{1/2} \cdot \int_{-\infty}^{\infty} n^4 \cdot \exp[-n^2/2v^2]\ dn \qquad \text{(A-7)}$$

Since the integrand is an even function of n, equation (A-7) may be written as

$$E\{ Z \} = 2 \cdot [1/(2\pi)v^2]^{1/2} \cdot \int_{0}^{\infty} n^4 \cdot \exp[-n^2/2v^2]\ dn \qquad \text{(A-8)}$$

Equation (A-8) is similiar to a standard form given in the CRC Standard Mathematical Tables as (Hodgman,1959:313)

$$\int_0^{\infty} x^{2n} \cdot \exp[-ax^2]\, dx = \frac{1 \cdot 3 \cdot 5 \cdots (2n-1) \cdot (\pi/a)^{1/2}}{2^{n+1} \cdot a^n} \qquad \text{(A-9)}$$

In this case n = 2 and $a = 1/2v^2$. Hence,

$$E\{ Z \} = 2[1/(2\pi)]^{1/2} \cdot (3/8v) \cdot (1/4v^4) \cdot [\pi/(1/2v^2)]^{1/2} \qquad \text{(A-10)}$$

After simplification,

$$E\{ Z \} = E\{n^4\} = 3v^4 \qquad \text{(A-11)}$$

In a similiar fashion, $E\{ Z^2\}$ is found to be

$$E\{ Z^2 \} = E\{n^8\} = 105 \cdot v^8 \qquad \text{(A-12)}$$

The variance of z is now calculated using the formula given above

$$\begin{aligned} Var\{ Z \} &= E\{ Z^2 \} - E^2\{ Z \} \\ &= 105v^8 - 9v^8 \\ &= 96(v^2)^4 \qquad \text{(A-13)} \end{aligned}$$

Equation (A-13) is the variance of the random process at the output of a fourth law device when the input is zero mean gaussian noise with a variance of v^2.

Next, the signal power at the output of the fourth law device will be calculated.

The signal s(t) is assumed to be of the form

$$s(t) = A\cos(w_c t + \phi) \tag{A-14}$$

This signal is applied to the input of the fourth law device. The output is calculated to be equal to

$$(A^4 / 4) \cdot [\ 1 + 2\cos(2wt) + 1/2 + (1/2)\cos(4wt)] \tag{A-15}$$

For the purposes of this thesis, only the signal component at four times the carrier frequency is required. This signal component is

$$(A^4 / 8) \cdot \cos(4wt) \tag{A-16}$$

The power in this component is calculated using the formula for sinusoidal signals, as before. In this case

$$P = (A^8 / 64\ ) \cdot (1/2) = A^8 / 128 \tag{A-17}$$

This is the power at the output of the fourth law device due to the signal input. Using the result from the above calculation, the output signal to noise ratio is given as

$$SNR_o = A^8 / 128 \cdot 96(v^2)^4 \tag{A-18}$$

The input to output SNR relationship is formed by taking the ratio of the output to input signal to noise ratios. This gives

$$SNR_o/SNR_{in} = [(A^2)^4/128 \cdot 96(v^2)^4\ ] / [A^2/2v^2] \tag{A-19}$$

After simplification and the rearrangement of terms, (A-19) can be written as

$$SNR_o/SNR_{in} = (A^2 / 2v^2)^3 / 768 \quad \text{(A-20)}$$

This is equivalent to

$$SNR_o/SNR_{in} = (SNR_{in})^3 / 768 \quad \text{(A-21)}$$

Hence, the output SNR is given in terms of the input SNR as

$$SNR_o = (SNR_{in})^4 / 768 \quad \text{(A-22)}$$

Appendix B: Program Listings

This appendix presents the listings of the programs used to perform the operations decsribed in Chapter IV. The listings begin on the following page.

```
C       THIS PROGRAM GENERATES OOK DATA PLUS NOISE


        PROGRAM    OOKGEN

C       DATE       28 OCT 1987

        BYTE                   PNBUF(256) ! BUFFER OF BITS OF PNCODE
        REAL                   CARBUF(400)          ! BUFFER OF CARRIER POINTS
        REAL                   RAWBUF(400)          ! BUFFER OF MODULATED CARRIER
        REAL                   NZBUFF(400)          ! BUFFER OF NOISE POINTS
        CHARACTER*32           FNAME
        CHARACTER  DUM                   ! DUMMY

C---    SOME USEFUL NUMBERS

        PI2 = 6.283185307
        FSAMP = 1000000.

C---    ENTER FREQUENCY OF CARRIER

        WRITE(6,390)
 390    FORMAT(2X,'ENTER CARRIER FREQUENCY: ',$)
        READ(6,391) FREQ
 391    FORMAT(G)

C---    ENTER BIT RATE

        WRITE(6,15)
 15     FORMAT(2X,'ENTER BIT RATE: ',$)
        READ(6,16)BITRAT
 16     FORMAT(G)

C---    CALCULATE NUMBER OF SAMPLES PER BIT

        NSMPBT = 1000000./BITRAT

C---    GET SOME FAKE BITS FROM THE PNCODE


        CALL PNREAD(PNBUF,NBITS)

C---    OPEN OUTPUT FILE

        OPEN( UNIT = 13,
        9     NAME = 'OOK.DAT',
        9     STATUS = 'NEW',
        9     ACCESS = 'SEQUENTIAL')

C---    THIS  NEXT STUFF IS FOR A MATRIXX FILE

        WRITE(13,55)
 55     FORMAT('Y = [')

C---    GET SOME CARRIER POINTS TO MULTIPLY WITH THE DATA

        OPEN (UNIT = 14,
        9     NAME = 'CARRIER.DAT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')

C---    NOW OPEN NOISE FILE, SINCE I'LL NEED IT LATER

        WRITE(6,134)
 134    FORMAT(2X'ENTER NAME OF NOISE FILE: ',$)
        READ(6,135)FNAME
 135    FORMAT(A)

        OPEN (UNIT = 15,
        9     NAME = FNAME,
        9     STATUS = 'OLD',
```

```
      9     ACCESS = 'SEQUENTIAL')

            READ(15,136)DUM                    ! DUMP MATRIXX OPENING
 136        FORMAT(A)
C---        NOW DETERMINE AMPLITUDE OF CARRIER BASED UPON DESIRED SNR
C---        UNITY VARIANCE GAUSSIAN NOISE

            WRITE(6,707)
 707        FORMAT(2X,'ENTER DESIRED SNR ( dB) : ',$)
            READ(6,708)SNR
 708        FORMAT(G)

            AMP = SQRT(2. * 10.0 ** (SNR/10.) )

 73         FORMAT(G)

C---        MULTIPLY THE CARRIER POINTS BY THE DATA, ADD NOISE AND
C           THEN WRITE TO OUTPUT FILE

            ICNT = 0

            DO JJ = 1,NBITS

               DO KK = 1,NSMPBT
                  READ(14,73) CARBUF(KK)
                  READ(15,10) NZBUFF(KK)
                  RAWBUF(KK) = AMP *  CARBUF(KK) * PNBUF(JJ) + NZBUFF(KK)
                  WRITE(13,10) RAWBUF(KK)
               END DO

            END DO


 10         FORMAT(G)


C---        WRITE MATRIXX EOF

            WRITE(13,56)
 56         FORMAT(']')

            CLOSE(13)
            CLOSE(14)
            CLOSE(15)

            STOP
            END
```

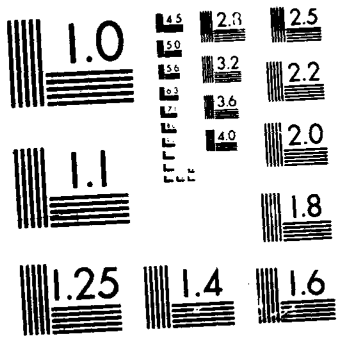
```
C       THIS PROGRAM GENERATES BPSK

        PROGRAM    BPSK

C       DATE       28 OCT 1987

        BYTE       PNBUF(256)          ! BUFFER OF BITS OF PNCODE
        REAL       CARBUF(400)                   ! BUFFER OF CARRIER POINTS
        REAL       RAWBUF(400)                   ! BUFFER OF MODULATED CARRIER


C---    SOME USEFUL NUMBERS

        PI2 = 6.283185307
        FREQ  =  100000.
        FSAMP = 1000000.

C---    ENTER BIT RATE

        WRITE(6,15)
 15     FORMAT(2X,'ENTER BIT RATE: ',$)
        READ(6,16)BITRAT
 16     FORMAT(G)

C---    CALCULATE NUMBER OF SAMPLES PER BIT

        NSMPBT = 1000000./BITRAT

C---    GET SOME FAKE BITS FROM THE PNCODE

        CALL PNREAD(PNBUF,NBITS)

C---    OPEN OUTPUT FILE

        OPEN( UNIT = 13,
        9     NAME = 'BPSK.DAT',
        9     STATUS = 'NEW',
        9     ACCESS = 'SEQUENTIAL')

C---    THIS  NEXT STUFF IS FOR A MATRIXX FILE

        WRITE(13,55)
 55     FORMAT('Y = [')

C---    GET SOME CARRIER POINTS TO MULTIPLY WITH THE DATA

        OPEN (UNIT = 14,
        9     NAME = 'CARRIER.DAT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')


 73     FORMAT(G)

C---    MULTIPLY THE CARRIER POINTS BY THE DATA, THEN WRITE TO OUTPUT FILE
C---    BPSK FORMED BY CHANGING 1/0 DATA TO (+/-) 1 DATA

        ICNT = 0

        DO JJ = 1,NBITS

           DO KK = 1,NSMPBT
              READ(14,73) CARBUF(KK)
              RAWBUF(KK) = CARBUF(KK) * ( (PNBUF(JJ)-.5) * 2.)
              WRITE(13,10) RAWBUF(KK)
           END DO

        END DO


 10     FORMAT(G)
 13     FORMAT(2X,'PNBUFF IS: ',I)
```

```
      WRITE(13,56)
56    FORMAT(']')

      CLOSE(13)


      STOP
      END
```

```
C       THIS PROGRAM GENERATES QPSK  DATA PLUS NOISE


        PROGRAM     QPSKGEN

C       DATE        28 OCT 1987

        BYTE_                   PNBUF(256) ! BUFFER OF BITS OF PNCODE
        REAL_                   CARBUF(400)          ! BUFFER OF CARRIER POINTS
        REAL_                   RAWBUF(400)          ! BUFFER OF MODULATED CARRIER
        REAL                    NZBUFF(400)          ! BUFFER OF NOISE POINTS
        BYTE                    DATBUF(128)          ! BUILD QUATS FROM BINARY DATA
        REAL                    EBUF(22)   ! BUFFER TO HOLD EVEN NUMBERED DATA BITS
        REAL                    OBUF(22)   ! BUFFER TO HOLD ODD NUMBERED DATA BITS
        CHARACTER*32            FNAME
        CHARACTER  DUM                     ! DUMMY

C---    SOME USEFUL NUMBERS

        PI2 = 6.283185307
        FSAMP = 1000000.

C---    ENTER FREQUENCY OF CARRIER

        WRITE(6,390)
 390    FORMAT(2X,'ENTER CARRIER FREQUENCY: ',$)
        READ(6,391) FREQ
 391    FORMAT(G)

C---    ENTER BIT RATE

        WRITE(6,15)
 15     FORMAT(2X,'ENTER SYMBOL RATE: ',$)
        READ(6,16)SYMRAT
 16     FORMAT(G)

C---    CALCULATE NUMBER OF SAMPLES PER BIT

        NSMPBT = 1000000./SYMRAT

C---    GET SOME FAKE BITS FROM THE PNCODE


        CALL PNREAD(PNBUF,NBITS)

C---    OPEN OUTPUT FILE

        OPEN( UNIT = 13,
        9     NAME = 'QPSK.DAT',
        9     STATUS = 'NEW',
        9     ACCESS = 'SEQUENTIAL')

C---    THIS  NEXT STUFF IS FOR A MATRIXX FILE

        WRITE(13,55)
 55     FORMAT('Y = [')

C---    GET SOME CARRIER POINTS TO MULTIPLY WITH THE DATA


C---    NOW OPEN NOISE FILE, SINCE I'LL NEED IT LATER

        WRITE(6,134)
 134    FORMAT(2X'ENTER NAME OF NOISE FILE: ',$)
        READ(6,135)FNAME
 135    FORMAT(A)

        OPEN (UNIT = 15,
        9     NAME = FNAME,
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')
```

AD-A194 623 AUTOMATIC CLASSIFICATION OF DIGITALLY MODULATED SIGNALS (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING M P DESIMIO DEC 87 2/2

UNCLASSIFIED AFIT/GE/ENG/87D-15 F/G 12/9 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

```
      READ(15,136)DUM                   ! DUMP MATRIXX OPENING
 136  FORMAT(A)

C---  NOW DETERMINE AMPLITUDE OF CARRIER BASED UPON DESIRED SNR
C---  UNITY VARIANCE GAUSSIAN NOISE

      WRITE(6,707)
 707  FORMAT(2X,'ENTER DESIRED SNR ( dB) : ',$)
      READ(6,708)SNR
 708  FORMAT(G)

      AMP = SQRT(2. * 10.0 ** (SNR/10.) )

 73   FORMAT(G)



C---  BUILD LOOK AT 2 BITS OF PNBUF TO DETERMINE HOW MUCH PHASE
C---  TO ADD TO THE COSINE TO GET THE PROPER QPSK ACTION


C---  SEPARATE EVEN AND ODD BITS

      DO KK = 1,NBITS/2
         EBUF(KK) = PNBUF(2*KK)
         OBUF(KK) = PNBUF(2*KK - 1)
      END DO


C---  SOME USEFUL NUMBERS

      PI = 3.1415926
      DELTAT = 1./FSAMP

C---  DEFINE PHASE SHIFTS

      PHI1 = PI/4.
      PHI2 = 3.*PI/4.
      PHI3 = 5.*PI/4.
      PHI4 = 7.*PI/4.

C---  CHOOSE APPROPRIATE PHASE SHIFT

      DO KK = 1,NBITS/2
         IF( (EBUF(KK) .EQ. 0) .AND. (OBUF(KK) .EQ. 0) ) PHI = PHI1
         IF( (EBUF(KK) .EQ. 0) .AND. (OBUF(KK) .EQ. 1) ) PHI = PHI2
         IF( (EBUF(KK) .EQ. 1) .AND. (OBUF(KK) .EQ. 0) ) PHI = PHI3
         IF( (EBUF(KK) .EQ. 1) .AND. (OBUF(KK) .EQ. 1) ) PHI = PHI4

C---  GENERATE COS(wt + PHIn) + NOISE AND WRITE TO OUTPUT FILE

         DO JJ = 1,NSMPBT
            CARBUF(JJ) = COS(PI2*FREQ*FLOAT(JJ)*DELTAT + PHI)
            READ(15,10) NZBUFF(JJ)
            RAWBUF(JJ) = AMP * CARBUF(JJ) + NZBUFF(JJ)
            WRITE(13,10) RAWBUF(JJ)
         END DO

      END DO

 10   FORMAT(G)


C---  WRITE MATRIXX EOF

      WRITE(13,56)
 56   FORMAT(']')

      CLOSE(13)
      CLOSE(15)
```

```
STOP
END
```

```
C       THIS PROGRAM GENERATES FSK  DATA PLUS NOISE


        PROGRAM    FSKGEN

C       DATE       28 OCT 1987

        BYTE                PNBUF(256)  ! BUFFER OF BITS OF PNCODE
        REAL                CARBUF(400)          ! BUFFER OF CARRIER POINTS
        REAL                CARBUF2(400)         ! BUFFER FOR OTHER CARRIER
        REAL                RAWBUF(400)          ! BUFFER OF MODULATED CARRIER
        REAL                NZBUFF(400)          ! BUFFER OF NOISE POINTS
        REAL                RBUF(256)   ! REAL BITS
        CHARACTER*32        FNAME
        CHARACTER  DUM                  ! DUMMY

C---    SOME USEFUL NUMBERS

        PI2 = 6.283185307
        FSAMP = 1000000.


C---    ENTER BIT RATE

        WRITE(6,15)
 15     FORMAT(2X,'ENTER BIT RATE: ',$)
        READ(6,16)BITRAT
 16     FORMAT(G)

C---    CALCULATE NUMBER OF SAMPLES PER BIT

        NSMPBT = 1000000./BITRAT

C---    GET SOME FAKE BITS FROM THE PNCODE


        CALL PNREAD(PNBUF,NBITS)
        WRITE(6,166)NBITS
 166    FORMAT(2X,I)

C---    OPEN OUTPUT FILE

        OPEN( UNIT = 13,
        9     NAME = 'FSK.DAT',
        9     STATUS = 'NEW',
        9     ACCESS = 'SEQUENTIAL')

C---    THIS  NEXT STUFF IS FOR A MATRIXX FILE

        WRITE(13,55)
 55     FORMAT('Y = [')

C---    GET SOME CARRIER POINTS TO MULTIPLY WITH THE DATA

        OPEN (UNIT = 14,
        9     NAME = 'CARRIER.DAT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')

        OPEN (UNIT = 24,
        9     NAME = 'CARRIER2.DAT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')


C---    NOW OPEN NOISE FILE, SINCE I'LL NEED IT LATER

        WRITE(6,134)
 134    FORMAT(2X'ENTER NAME OF NOISE FILE: ',$)
        READ(6,135)FNAME
```

```
 135    FORMAT(A)

        OPEN (UNIT = 15,
        9     NAME = FNAME,
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')

        READ(15,136)DUM                        ! DUMP MATRIXX OPENING
 136    FORMAT(A)

C---    NOW DETERMINE AMPLITUDE OF CARRIERS BASED UPON DESIRED SNR
C---    UNITY VARIANCE GAUSSIAN NOISE

        WRITE(6,707)
 707    FORMAT(2X,'ENTER DESIRED SNR ( dB) : ',$)
        READ(6,708)SNR
 708    FORMAT(G)

        AMP = SQRT(2. * 10.0 ** (SNR/10.) )

 73     FORMAT(G)

C---    MULTIPLY THE CARRIER POINTS BY THE DATA, ADD NOISE


        DO JJ = 1,NBITS

           DO KK = 1,NSMPBT
              READ(14,73) CARBUF(KK)
              READ(24,73) CARBUF2(KK)
              READ(15,10) NZBUFF(KK)

              IF(PNBUF(JJ) .EQ. 1) THEN
                 RAWBUF(KK) = AMP*CARBUF(KK) + NZBUFF(KK)
              ENDIF

              IF(PNBUF(JJ) .EQ. 0) THEN
                 RAWBUF(KK) = AMP*CARBUF2(KK) + NZBUFF(KK)
              ENDIF

              WRITE(13,10) RAWBUF(KK)
           END DO

        END DO

 155    FORMAT(2X,'JJ: ',I,5X,'KK: ',I,10X,'RAWBUF: ',G)


 10     FORMAT(G)


C---    WRITE MATRIXX EOF

        WRITE(13,56)
 56     FORMAT(']')

        CLOSE(13)
        CLOSE(14)
        CLOSE(15)
        CLOSE(24)
        STOP
        END
```

```fortran
C       THIS PROGRAM GENERATES GAUSSIAN NOISE OF DESIRED VARIANCE
C       THE ROUTINE USED TO GENERATE SAMPLES FROM A UNIFORMLY DIST-
C       RIBUTED RANDOM PROCESS IS FROM WIDROW AND STEARNS.

        PROGRAM              GNOISE

        REAL                 YBARBUF(8800)
        REAL                 ACCUM(8800)
        REAL_                R(8800)
        REAL-                NEWVAR
        REAL-                GAUSS(8800)
        CHARACTER*32         FNAME

        WRITE(6,50)
 50     FORMAT(2X,'ENTER NAME OF OUTPUT FILE: ',$)
        READ (6,51) FNAME
 51     FORMAT(A)

        OPEN(UNIT = 3,
        9    NAME = FNAME,
        9    STATUS = 'NEW',
        9    ACCESS = 'SEQUENTIAL')

        WRITE(3,55)
 55     FORMAT('Y=[')

        WRITE(6,60)
 60     FORMAT(2X,'ENTER DESIRED VARIANCE: ',$)
        READ(6,61) NEWVAR
 61     FORMAT(G)

        WRITE(6,52)
52      FORMAT(2X,'ENTER SEED FOR RANDOM NUMBER GENERATOR: ',$)
        READ(6,53) K
 53     FORMAT(I)

C---    ADD 50 RANDOM VECTORS SO ELEMENTS WILL BE APPROXIMATELY GAUSSIAN
C---    DISTRIBUTED RANDOM VARIABLES


        DO JJ = 1,50

           DO KK = 1,8800
              R(KK) = RANDOM(K) - .5
           END DO

           DO KK = 1,8800
              ACCUM(KK) = ACCUM(KK) + R(KK)
           END DO

        END DO

C---    NORMALIZE TO STANDARD NORMAL: MEAN IS NOW ZERO AND VARIANCE IS NOW 1/12

        DO KK = 1,8800
           YBARBUF(KK) = ACCUM(KK)/50.
        END DO

C---    NOW GET THE DESIRED VARIANCE

        DO KK = 1,8800
           GAUSS(KK) = SQRT(NEWVAR) * SQRT(50.)*YBARBUF(KK)/SQRT(1./12.)
        END DO


C---    WRITE OUT GAUSSIAN VECTOR TO FILE

        DO KK = 1,8800
           WRITE(3,10)GAUSS(KK)
        END DO

 10     FORMAT(G)
```

```
      WRITE(3,59)
59    FORMAT(']')

      CLOSE(3)

      STOP
      END

      FUNCTION RANDOM(I)

      I = 2045*I+1
      I = I -(I/1048576) * 1048576
      RANDOM = FLOAT(I+1)/1048577.0
      RETURN
      END
```

```
C       THIS PROGRAM CALCULATES THE ENVELOPE OF SIGNALS BY USING THE
C       HILBERT TRANSFORM TO FIND THE QUADRATURE COMPONENT.  THEN THE
C       STANDARD FORMULA OF SQRT(I**2 + Q**2) IS USED TO FIND THE ENVELOPE.

        PROGRAM    ENVELOPE

        REAL                    BUFF(8192) ! DATA READ FROM INPUT FILE
        REAL                    RX(8192)   ! REAL BUFFER FROM COMPLEX
        REAL                    ENV(8192)  ! BUFFER TO HOLD REAL ANSWER
        COMPLEX                 X(8192)              ! BUFFER OF COMPLEX INPUT DATA
        COMPLEX                 POSHIL               ! CONSTANT EQUAL TO -j
        COMPLEX                 NEGHIL               ! CONSTANT EQUAL TO +j
        CHARACTER*32            FNAME                ! NAME OF INPUT FILE
        CHARACTER*32            FNAME55              ! NAME OF OUTPUT FILE
        CHARACTER  DUM                     ! DUMMY

        POSHIL = CMPLX( 0., -1.)
        NEGHIL = CMPLX( 0., 1.)

C---    GET SIGNAL DATA

        WRITE(6,1)
 1      FORMAT(2X,'ENTER NAME OF INPUT FILE: ',$)
        READ(6,2) FNAME
 2      FORMAT(A)



        OPEN (UNIT = 3,
        9      NAME = FNAME,
        9      STATUS = 'OLD',
        9      ACCESS = 'SEQUENTIAL')

C---    DUMP MATRIXX BEGINNING

        READ(3,55)DUM
 55     FORMAT(A)
C---    GO

        DO KK = 1,8192
           READ(3,10) BUFF(KK)
           X(KK) = BUFF(KK)
        END DO

10      FORMAT(G)

C---    CLOSE INPUT FILE

        CLOSE(3)

C---    THIS PROGRAM SET UP FOR 4096 POINT DATA SEGMENT AND FFT

        N = 8192
        INV = 0                          ! FORWARD FFT

        CALL FFT(X,N,INV)

C---    THIS IS THE HILBERT TRANSFORM PART

        DO KK = 1,4096
           X(KK) = X(KK) * POSHIL
        END DO

        DO KK = 4097,8192
           X(KK) = X(KK) * NEGHIL
        END DO

C---    THAT'S THAT.   NOW INVERSE TRANSFORM

        INV = 1                          ! INVERSE FFT

        CALL FFT(X,N,INV)
```

```
C---   USE INPHASE AND QUADRATURE (HILBERT(X)) TO GET ENVELOPE

       DO KK = 1,8192
          RX(KK) = X(KK)
          ENV(KK) = SQRT(BUFF(KK)**2 + RX(KK)**2)
       END DO

C---   OPEN OUTPUT FILE

       WRITE(6,1001)
 1001  FORMAT(2X,'ENTER NAME OF OUTPUT FILE: ',$)
       READ(6,1002) FNAME55
 1002  FORMAT(A)



       OPEN( UNIT = 11,
       9     NAME = FNAME55,
       9     STATUS = 'NEW',
       9     ACCESS = 'SEQUENTIAL')

C---   WRITE DATA TO OUTPUT FILE

C---   MATRIXX OPENING STUFF

       WRITE(11,3)
  3    FORMAT('Y = [')

       DO KK = 1,8192
          WRITE(11,34) ENV(KK)
       END DO

  34   FORMAT(G)

C---   MATRIXX STUFF

       WRITE(11,35)
  35   FORMAT(']')

C---   CLOSE UP AND SHUT DOWN

       CLOSE(11)

       STOP
       END
```

```
C       THIS PROGRAM CALCULATES THE MEAN AND VARIANCE OF FILES

        PROGRAM    STATS

        REAL                  X(8800)                       ! INPUT BUFFER
        CHARACTER  DUM                           ! DUMMY
        CHARACTER*32          FNAME                         ! INPUT FILENAME
        BYTE                  MAT

C---    OPEN INPUT FILE

        WRITE(6,101)
 101    FORMAT(2X,'ENTER INPUT FILE: ',$)
        READ(6,102) FNAME
 102    FORMAT(A)

        WRITE(6,201)
 201    FORMAT(2X,'IS THIS A MATRIXX FILE [Y/N]: ',$)
        READ(6,202) MAT
 202    FORMAT(A1)

        WRITE(6,103)
 103    FORMAT(2X,'ENTER NUMBER OF DATA POINTS IN INPUT FILE: ',$)
        READ(6,104)NPNT
 104    FORMAT(I)

        OPEN(UNIT = 3,
        9    NAME = FNAME,
        9    STATUS = 'OLD',
        9    ACCESS = 'SEQUENTIAL')

C---    NIX MATRIXX BEGINNING TO FILE IF NECESSARY

        IF( MAT .EQ. 'Y') THEN
           READ(3,1)DUM
        ENDIF
 1      FORMAT(A)

C---    DO THE REAL READ

        DO KK = 1,NPNT
          READ(3,3) X(KK)
        END DO

 3      FORMAT(G)

C---    CLOSE THE INPUT FILE

        CLOSE (3)

C---    CALCULATE THE SAMPLE MEAN

        SUM = 0.0

        DO KK = 1,NPNT
           SUM = SUM + X(KK)
        END DO

        SAMEAN = SUM/FLOAT(NPNT)

C---    CALCULATE THE SAMPLE VARIANCE

        SUM = 0.0

        DO KK = 1,NPNT
           SUM = SUM + ( X(KK) - SAMEAN)**2
        END DO

        SAMVAR = SUM/FLOAT(NPNT)

C---    NOW WRITE RESULTS TO OUTPUT FILE
```

```
C       OPEN(UNIT = 4,
C       9     NAME = 'STATS.DAT',
C       9     STATUS = 'NEW',
C       9     ACCESS = 'SEQUENTIAL')

        WRITE(6,5) SAMEAN,SAMVAR
 5      FORMAT(2X,'SAMPLE MEAN: ',G,10X,'SAMPLE VARIANCE: ',G)

C       CLOSE(4)

        STOP
        END
```

```
C       THIS PROGRAM CALCULATES THE SPECTRUM OF N POINTS
C       THIS PROCESS IS REPEATED M TIMES, AND THE M SPECTRA
C       ARE AVERAGED.

        PROGRAM    SPECAVG

        CHARACTER*32           FNAME                   ! FILENAME
        REAL                   RX(4096)    ! BUFFER OF REAL DATA POINTS
        REAL                   MAG(2048)   ! STORES MAGNITUDE SQUARED OF FFT RESULT
        COMPLEX                X(4096)                 ! COMPLEX BUFFER FOR FFT SUBROUTINE
        REAL                   ACCUM(4096)             ! ACCUMULATOR TO AVERAGE SPECTRA

        WRITE(6,1)
 1      FORMAT(2X,'ENTER FILENAME OF INPUT DATA: ',$)
        READ(6,2)FNAME
 2      FORMAT(A)

        WRITE(6,33) FNAME
 33     FORMAT(2X,'FILENAME OF INPUT DATA IS: ',A)

        OPEN(UNIT = 12,
        9    NAME = FNAME,
        9    STATUS = 'OLD',
        9    ACCESS = 'SEQUENTIAL')


        WRITE(6,219)
 219    FORMAT(2X,'ENTER NAME OF OUTPUT FILE: ',$)
        READ(6,220)FNAME
 220    FORMAT(A)

 101    OPEN(UNIT = 13,
        9    NAME = FNAME,
        9    STATUS = 'NEW',
        9    ACCESS = 'SEQUENTIAL')


C       WRITE(6,3)
C 3     FORMAT(2X,'ENTER NUMBER OF POINTS IN FFT: ',$)
C       READ(6,4) NFFTPT
C 4     FORMAT(I)

        NFFTPT = 4096

C       WRITE(6,34)
C 34    FORMAT(2X,'ENTER NUMBER OF SPECTRA TO CALCULATE AND AVERAGE: ',$)
C       READ(6,35)ITERLIM
C 35    FORMAT(I)

        ITERLIM = 2

C---    NIX MATRIXX STUFF

        READ(12,9)DUMMY
 9      FORMAT(A)

C---    NOW GET ON WITH THE REAL READ

        ITER = 0

 22     ITER = ITER + 1

        DO KK = 1,NFFTPT
           READ(12,10,ERR = 39)RX(KK)
           X(KK) = RX(KK)
        END DO
 10     FORMAT(G)

 39     CONTINUE


C---    CALL THE FFT SUBROUTINE
```

```
      INV = 0                                       ! IMPLIES FORWARD TRANSFORM
      N = NFFTPT
      CALL FFT(X,N,INV)

C---  NOW FIND THE MAGNITUDE SQUARED

      DO KK = 1,NFFTPT/2
         MAG(KK) = CABS( X(KK))**2
      END DO

C---  DEBUG STUFF

      DO KK = 1,10
         WRITE(6,335)KK,MAG(KK)
      END DO
 335  FORMAT(2X,'MAG(',I2,') = ',G)

C---  NOW ACCUMULATE THE SPECTRA

      DO KK = 1,NFFTPT/2
         ACCUM(KK) = ACCUM(KK) + MAG(KK)
      END DO

      IF (ITER .NE. ITERLIM)GOTO 22
C---  IF YOU GET HERE, YOU ARE FINISHED WITH THE INPUT FILE

      CLOSE(12)

C---  NOW WRITE THIS OUT TO A FILE


C---  MATRIXX FILE FORMAT STUFF

      WRITE(13,108)
 108  FORMAT('Y = [')

      DO KK = 1,NFFTPT/2
         ACCUM(KK) = ACCUM(KK)/FLOAT(ITERLIM)
         WRITE(13,21)ACCUM(KK)
      END DO

 21   FORMAT(G)

C---  MORE MATRIXX FORMAT

      WRITE(13,109)
 109  FORMAT(']')
      CLOSE (13)

      STOP
      END
```

```
C       THIS FFT SUBROUTINE COMES FROM THE BOOK DISCRETE TIME SIGNALS
C       AND SYSTEMS BY AHMED AND NATARAJAN, APPENDIX 4.1
C
C***********************************************************************
C
C       CALLING SEQUENCE
C                  CALL FFT(X,N,INV)
C
C       ARGUMENTS REQUIRED FROM THE CALLING ROUTINE
C
C                  X            - COMPLEX VECTOR TO BE TRANSFORMED
C                  N            - NUMBER OF POINTS TO BE TRANSFORMED
C                     (MUST BE A POWER OF 2)
C                  INV          - INV = 0 ==> FORWARD TRANSFORM
C                                 INV = 1 ==> INVERSE TRANSFORM
C
C       ARGUMENTS SUPPLIED TO THE CALLING ROUTINE
C
C                  X            - COMPLEX TRANSFORMED VECTOR
C                                 NOTE THAT THE TRANSFORMED VECTOR IS RETURNED
C                                 IN THE ORIGINAL TIME ARRAY OF POINTS
C***********************************************************************

        SUBROUTINE FFT(X,N,INV)
        COMPLEX X(1),W,T
        ITER = 0
        IREM = N
 10     IREM = IREM/2
        IF (IREM .EQ. 0) GOTO 20
        ITER = ITER + 1
        GOTO 10
 20     CONTINUE
        S = -1
        IF (INV .EQ. 1) S = 1
        NXP2 = N
        DO 50 IT = 1,ITER
        NXP = NXP2
        NXP2 = NXP/2
        WPWR = 3.1415926/FLOAT(NXP2)
        DO 40 M = 1,NXP2
        ARG = FLOAT(M-1)*WPWR
        W = CMPLX(COS(ARG),S*SIN(ARG))
        DO 40 MXP = NXP,N,NXP
        J1 = MXP-NXP+M
        J2 = J1 + NXP2
        T = X(J1)-X(J2)
        X(J1) = X(J1)+X(J2)
 40     X(J2) = T*W
 50     CONTINUE
        N2=N/2
        N1=N-1
        J=1
        DO 65 I=1,N1
        IF(I .GE. J) GOTO 55
        T=X(J)
        X(J) = X(I)
        X(I) = T
 55     K=N2
 60     IF(K .GE. J) GOTO 65
        J=J-K
        K=K/2
        GOTO 60
 65     J=J+K
        IF (INV .EQ. 1) GOTO 75
        DO 70 I=1,N
 70     X(I) = X(I)/FLOAT(N)
 75     CONTINUE
        RETURN
        END
```

```
C       THIS PROGRAM WILL DO SPECTRAL CORRELATION OF DATA SPECTRA WITH
C       A SINC SQUARED FUNCITON.  THE INPUT SPECTRA ARE TREATED AS WAVEFORMS
C       SO THE ENERGY OF EACH ARE NORMALIZED TO UNITY BEFORE  THE CORRELATIONS
C       ARE PERFORMED

        PROGRAM     SPECOR

        REAL        BUFF0(4096), BUFF1(4096), BUFF2(8192)
        REAL_       BUFF3(4096),MIS
        REAL        SYMRAT
        REAL        MAXVAL
        INTEGER     PEAKLC
        CHARACTER*32            FNAME
        BYTE                    DUM

C---    INPUT ACTUAL SYMBOL RATE OF DATA

C       WRITE(6,3000)
C 3000  FORMAT(2X,'ENTER BASEBAND SYMBOL RATE IN HZ: ',$)
C       READ(6,3001)SYMRAT
C 3001  FORMAT(G)

        SYMRAT = 500.

C---    SOME USEFUL NUMBERS...INCLUDE FFT SIZE

C       WRITE(6,4000)
C 4000  FORMAT(2X,'ENTER BIN SIZE OF FFT: ',$)
C       READ(6,4001)DELFRQ
C 4001  FORMAT(G)

        DELFRQ = 244.140625

        PI2 = 2.0 * 3.1415926

C---    NOW GENERATE THE BASEBAND SINC SQUARED
C---    NOTE THAT THE DC COMPONENT OF THIS SINC SQUARED
C---    IS AT BUFFER LOCATION 1024

        DELTAT = 1./SYMRAT

        DO JJ = 1,2048
           TMP = PI2*DELTAT*DELFRQ*(FLOATJ(JJ) -1023.999)
           BUFF0(JJ) = (SIN(TMP)/TMP)**2
        END DO

C--     CALCULATE THE ENERGY IN THE BASEBAND SINC SQUARED

        ESUM = 0.0

        DO JJ = 1,2048
           ESUM = BUFF0(JJ)**2 + ESUM
        ENDDO

C---    NOW NORMAIZE SUCH THAT ENERGY OF WAVEFORM IS ONE

        DO JJ = 1,2048
           BUFF0(JJ) = BUFF0(JJ)/SQRT(ESUM)
        ENDDO

C---    READ IN THE DATA FILE

        WRITE(6,181)
 181    FORMAT(2X,'ENTER INPUT FILE: ',$)
        READ(6,183) FNAME
 183    FORMAT(A)

        OPEN(UNIT = 11,
        9       NAME = FNAME,
        9       STATUS = 'OLD',
        9       ACCESS = 'SEQUENTIAL')

C---    GET OUTPUT FILE NAME
```

```
        WRITE(6,281)
  281   FORMAT(2X,'ENTER OUTPUT FILENAME: ',$)
        READ(6,183)FNAME

C---    NIX MATRIXX STUFF

        READ(11,173)DUM
  173   FORMAT(A)

        DO KK = 1,2048
           READ(11,177)BUFF1(KK)
        END DO
  177   FORMAT(G)

        CLOSE(11)
C---    NIX THE DC RESPONSE BEFORE THE CORRELATIONS

        BUFF1(1) = 0.

C---    CALCULATE THE ENERGY IN THE SPECTRA WAVEFORM

        ESUM = 0.

        DO KK = 1,2048
           ESUM = BUFF1(KK)**2 + ESUM
        END DO

C---    NOW NORMALIZE

        DO KK = 1,2048
           BUFF1(KK) = BUFF1(KK)/SQRT(ESUM)
        END DO

C---    TAKE CARE OF OFFSET OF BASEBAND SINC BEFORE CORRELATION
C---    THIS IS ACCOMPLISHED BY PUSHING THE SPECTRUM OF THE BANDPASS
C---    SIGNAL OUT 1023 POINTS

        DO N = 1,3072
           NOFFSET = N - 1023

           IF( NOFFSET .LT. 1) THEN
              BUFF3(N) = 0.0
           ELSE
              BUFF3(N) = BUFF1(N-1023)
           ENDIF

        END DO


C---    HERE COMES THE CORRELATION

        DO  N = 1,2048
           TEMP = 0.0
           DO KK = 1,2048
              SUM = BUFF0(KK)*BUFF3(KK+N-1)
              TEMP = TEMP + SUM
           END DO
           BUFF2(N) = TEMP
        END DO



C---    NOW WRITE OUT THE OUTPUT FILE

        OPEN(UNIT = 3,
        9    FILE = FNAME,
        9    STATUS = 'NEW',
        9    ACCESS = 'SEQUENTIAL')

C---    MATRIXX FILE FORMAT

        WRITE(3,498)
```

```
 498    FORMAT('Y = [')

        DO KK = 1,2048
           WRITE(3,501)BUFF2(KK)
        ENDDO
 501    FORMAT(G)

C---    MATRIXX FILE FORMAT

        WRITE(3,499)
 499    FORMAT(']')

C---    CLOSE OUTPUT FILE

        CLOSE(3)

C--     NOW FOR MY INFORMATION, GIVE PEAK CORRELATION VALUE AND LOCATION

        DO KK = 1,2048
           IF(BUFF2(KK) .GT. MAXVAL) THEN
              MAXVAL = BUFF2(KK)
              PEAKLC = KK
           ENDIF
        ENDDO
        WRITE(6,555)MAXVAL,PEAKLC
 555    FORMAT(2X,'MAX CORR VAL IS: ',F15.10,10X,'PEAK LOC: ',I)

        STOP
        END
```

```
C       THIS PROGRAM DESIGNED TO BE USED WITH SPECOR FILES ONLY
C       AND FINDS THE TWO LARGEST NUMBERS IN THE CORRELATION AND
C       SAVES THEIR LOCATIONS

        PROGRAM                 BIGVALS

        CHARACTER*32            FNAME,DUM
        BYTE                    MATFLG
        REAL                    UMAX
        REAL                    BMAX
        INTEGER                 UMAXLOC
        INTEGER                 BMAXLOC
        INTEGER                 TWIDDLE
        REAL                    BUFF(2048)

C---    GET ON WITH IT

        WRITE(6,15)
 15     FORMAT(2X,'ENTER FILENAME: ',$)
        READ(6,16) FNAME
 16     FORMAT(A)


        OPEN(UNIT = 3,
        9    NAME = FNAME,
        9    STATUS = 'OLD',
        9    ACCESS = 'SEQUENTIAL')


        WRITE(6,25)
 25     FORMAT(2X,'IS THIS A MATRIXX TYPE FILE [Y/N]: ',$)
        READ(6,26) MATFLG
 26     FORMAT(A)

        IF (MATFLG .EQ. 'Y') READ(3,26)DUM

        DO KK = 1,2048
           READ(3,20)BUFF(KK)
        END DO

C---    GET RID OF LARGE DC RESPONSE BY NIXING LOW FREQUENCY VALUES


        UMAX = -500000.
        BMAX = -500000.

        DO KK = 1,2048
           IF(BUFF(KK) .GT. UMAX) THEN
              UMAX = BUFF(KK)
              ILOC = KK
           ENDIF
        END DO

C---    ZERO OUT POINTS NEAR THE BIGGEST POINT

        IF(ILOC .LT. 8) STOP' MAXLOC IS REALLY SMALL'

        DO KK = ILOC-8,ILOC+8
           BUFF(KK) = 0.0
        END DO

        UMAX1 = -5000.
        DO KK = 1,2048
           IF(BUFF(KK) .GT. UMAX1) THEN
              UMAX1 = BUFF(KK)
              ILOC1 = KK
           ENDIF
        END DO

 101    FORMAT(2X,'NUMBER OF POINTS IN FILE IS: ',I)
 20     FORMAT(G)
```

```
      CLOSE(3)

      WRITE(6,50)UMAX,ILOC
      WRITE(6,51)UMAX1,ILOC1

50    FORMAT(2X,'BIGGEST IS: ',G,10X,'AT LOCATION: ',I)
51    FORMAT(2X,'SECOND BIGGEST IS: ',G,10X,'AT LOCATION: ',I)

      STOP
      END
```

```
C       THIS PROGRAM DESIGNED TO BE USED WITH SPECOR FILES ONLY
C       AND IT SEARCHES FOR PEAKS IN VICINTIY OF 600 TO 1000 FFT BINS

        PROGRAM                 SVAL

        CHARACTER*32            FNAME,DUM
        BYTE                    MATFLG
        REAL                    UMAX
        REAL                    BMAX
        INTEGER                 UMAXLOC
        INTEGER                 BMAXLOC
        INTEGER                 TWIDDLE
        REAL                    BUFF(2048)

C---    GET ON WITH IT

 1001   WRITE(6,15)
  15    FORMAT(2X,'ENTER FILENAME: ',$)
        READ(6,16,END = 9999) FNAME
 16     FORMAT(A)


        OPEN(UNIT = 3,
        9    NAME = FNAME,
        9    STATUS = 'OLD',
        9    ACCESS = 'SEQUENTIAL')


        WRITE(6,25)
 25     FORMAT(2X,'IS THIS A MATRIXX TYPE FILE [Y/N]: ',$)
        READ(6,26) MATFLG
 26     FORMAT(A)

        IF (MATFLG .EQ. 'Y') READ(3,26)DUM

        DO KK = 1,2048
           READ(3,20)BUFF(KK)
        END DO



        UMAX = -500000.
        BMAX = -500000.
C---    CHECK POINTS ONLY NEAR WHERE EXPECTED

        DO KK = 600,1000
           IF(BUFF(KK) .GT. UMAX) THEN
              UMAX = BUFF(KK)
              ILOC = KK
           ENDIF
        END DO

C---    ZERO OUT POINTS NEAR THE BIGGEST POINT

        DO KK = ILOC-8,ILOC+8
           BUFF(KK) = 0.0
        END DO

        UMAX1 = -5000.
        DO KK = 1,2048
           IF(BUFF(KK) .GT. UMAX1) THEN
              UMAX1 = BUFF(KK)
              ILOC1 = KK
           ENDIF
        END DO

 101    FORMAT(2X,'NUMBER OF POINTS IN FILE IS: ',I)
 20     FORMAT(G)

        CLOSE(3)
```

```
      WRITE(6,50)UMAX,ILOC
      WRITE(6,51)UMAX1,ILOC1

50    FORMAT(2X,'BIGGEST IS: ',G,10X,'AT LOCATION: ',I)
51    FORMAT(2X,'SECOND BIGGEST IS: ',G,10X,'AT LOCATION: ',I)

      GOTO 1001

9999  STOP
      END
```

```
C       THIS PROGRAM DESIGNED TO BE USED WITH SPECOR FILES ONLY

        PROGRAM                 QVAL

        CHARACTER*32            FNAME,DUM
        BYTE                    MATFLG
        REAL                    UMAX
        REAL                    BMAX
        INTEGER                 UMAXLOC
        INTEGER                 BMAXLOC
        INTEGER                 TWIDDLE
        REAL                    BUFF(2048)


C---    GET ON WITH IT

 1001   WRITE(6,15)
  15    FORMAT(2X,'ENTER FILENAME: ',$)
        READ(6,16,END = 9999) FNAME
  16    FORMAT(A)


        OPEN(UNIT = 3,
        9     NAME = FNAME,
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')


        WRITE(6,25)
  25    FORMAT(2X,'IS THIS A MATRIXX TYPE FILE [Y/N]: ',$)
        READ(6,26) MATFLG
  26    FORMAT(A)

        IF (MATFLG .EQ. 'Y') READ(3,26)DUM

        DO KK = 1,2048
           READ(3,20)BUFF(KK)
        END DO



        UMAX = -500000.
        BMAX = -500000.
C---    CHECK POINTS ONLY NEAR WHERE EXPECTED

        DO KK = 1400,1800
           IF(BUFF(KK) .GT. UMAX) THEN
              UMAX = BUFF(KK)
              ILOC = KK
           ENDIF
        END DO

C---    ZERO OUT POINTS NEAR THE BIGGEST POINT

        DO KK = ILOC-8,ILOC+8
           BUFF(KK) = 0.0
        END DO

        UMAX1 = -5000.
        DO KK = 1,2048
           IF(BUFF(KK) .GT. UMAX1) THEN
              UMAX1 = BUFF(KK)
              ILOC1 = KK
           ENDIF
        END DO

  101   FORMAT(2X,'NUMBER OF POINTS IN FILE IS: ',I)
  20    FORMAT(G)


        CLOSE(3)

        WRITE(6,50)UMAX,ILOC
```

```
      WRITE(6,51)UMAX1,ILOC1

50    FORMAT(2X,'BIGGEST IS: ',G,10X,'AT LOCATION: ',I)
51    FORMAT(2X,'SECOND BIGGEST IS: ',G,10X,'AT LOCATION: ',I)

      GOTO 1001

9999  STOP
      END
```

```
C       THIS PROGRAM IMPLEMENTS A MODIFIED LMS ALGORITHM BASED UPON A COM-
C       BINATION OF IDEAS FROM TOU AND GONZALEZ, LIPPMANN, AND TREICHLER
C       AND OTHERS

C       DATE 3 NOVEMBER 1987

        PROGRAM THELMS

        REAL        X1(10),  X2(10),  X3(10), X4(10) ! FEATURE VECTORS
        REAL        W1(10),  W2(10),  W3(10), W4(10) ! WEIGHT VECTORS
        REAL        W5(10)
        REAL        D1,D2,D3,D4,D5                    ! DESIRED OUTPUT VALUES
        REAL        E1,E2,E3,E4,E5                    ! DESIRED MINUS ACTUAL
        REAL        Y1,Y2,Y3,Y4,Y5                    ! ACTUAL OUTPUT VALUES
        REAL        MU                                ! GAIN PARAMETER: "TWIDDLE FACTOR"
        BYTE        AGNFLG
        BYTE        SKIP

C---    GIVE FEATURE VECTORS VALUES

        WRITE(6,800)
 800    FORMAT(2X,'ENTER NUMBER OF ELEMENTS IN FEATURE VECTORS: ',$)
        READ(6,801)NEL
 801    FORMAT(I)

 73     OPEN (UNIT = 3,
        9     NAME = 'OOK.FT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')

        DO KK = 1,NEL
           READ(3,810)X1(KK)
        END DO

        CLOSE(3)

        OPEN (UNIT = 3,
        9     NAME = 'BPSK.FT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')

        DO KK = 1,NEL
           READ(3,810)X2(KK)
        END DO

        CLOSE(3)

        OPEN (UNIT = 3,
        9     NAME = 'QPSK.FT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')

        DO KK = 1,NEL
           READ(3,810)X3(KK)
        END DO

        CLOSE(3)

        OPEN (UNIT = 3,
        9     NAME = 'FSK.FT',
        9     STATUS = 'OLD',
        9     ACCESS = 'SEQUENTIAL')

        DO KK = 1,NEL
           READ(3,810)X4(KK)
        END DO

        CLOSE(3)

 810    FORMAT(G)

C---    INITIALIZE WEIGHT VECTORS TO ZERO
```

```
 74     DO KK = 1,NEL
           W1(KK) = 0.0
           W2(KK) = 0.0
           W3(KK) = 0.0
           W4(KK) = 0.0
        END DO

C---    INITIALIZE GAIN CONSTANT

        WRITE(6,1)
  1     FORMAT(2X,'ENTER GAIN CONSTANT: ',$)
        READ(6,2)MU
  2     FORMAT(G)

C---    ENTER NUMBER OF DESIRED ITERATIONS

        WRITE(6,93)
 93     FORMAT(2X,'ENTER NUMBER OF ITERATIONS: ',$)
        READ(6,94)ITERLIM
 94     FORMAT(I)

C---    BEGIN ITERATIONS

        ITER = 0

C---    GET ITERATIONS REALTED TO INDEX OF CLASSES

 10     ITER = ITER + 1

        IVAL = IIFIX(AMOD(FLOAT(ITER),4. ))
        IF (IVAL .EQ. 0) IVAL = 4

C---    GET DESIRED OUTPUT VALUES FOR EACH ITERATION

        IF (IVAL .EQ. 1) THEN
           D1 = 1.
           D2 = 0.
           D3 = 0.
           D4 = 0.
        ENDIF

        IF (IVAL .EQ. 2) THEN
           D1 = 0.
           D2 = 1.
           D3 = 0.
           D4 = 0.
        ENDIF

        IF (IVAL .EQ. 3) THEN
           D1 = 0.
           D2 = 0.
           D3 = 1.
           D4 = 0.
        ENDIF

        IF (IVAL .EQ. 4) THEN
           D1 = 0.
           D2 = 0.
           D3 = 0.
           D4 = 1.
        ENDIF


C---    CALCULATE ACTUAL OUTPUT VALUES FOR EACH SET OF WEIGHTS

        Y1 = 0.
        Y2 = 0.
        Y3 = 0.
        Y4 = 0.

        IF (IVAL .EQ. 1) THEN
           DO KK = 1,NEL
```

```
            Y1 = W1(KK) * X1(KK) + Y1
            Y2 = W2(KK) * X1(KK) + Y2
            Y3 = W3(KK) * X1(KK) + Y3
            Y4 = W4(KK) * X1(KK) + Y4
         ENDDO
      ENDIF

      IF (IVAL .EQ. 2) THEN
         DO KK = 1,NEL
            Y1 = W1(KK) * X2(KK) + Y1
            Y2 = W2(KK) * X2(KK) + Y2
            Y3 = W3(KK) * X2(KK) + Y3
            Y4 = W4(KK) * X2(KK) + Y4
         ENDDO
      ENDIF

      IF (IVAL .EQ. 3) THEN
         DO KK = 1,NEL
            Y1 = W1(KK) * X3(KK) + Y1
            Y2 = W2(KK) * X3(KK) + Y2
            Y3 = W3(KK) * X3(KK) + Y3
            Y4 = W4(KK) * X3(KK) + Y4
         ENDDO
      ENDIF

      IF (IVAL .EQ. 4) THEN
         DO KK = 1,NEL
            Y1 = W1(KK) * X4(KK) + Y1
            Y2 = W2(KK) * X4(KK) + Y2
            Y3 = W3(KK) * X4(KK) + Y3
            Y4 = W4(KK) * X4(KK) + Y4
         ENDDO
      ENDIF

C---  CALCULATE ERRORS

      E1 = D1 - Y1
      E2 = D2 - Y2
      E3 = D3 - Y3
      E4 = D4 - Y4

C---  NOW DO THE UPDATES OF THE WEIGHT VECTORS

      IF(IVAL .EQ. 1) THEN
         DO KK = 1,NEL
            W1(KK) = W1(KK) + MU * E1 * X1(KK)
            W2(KK) = W2(KK) + MU * E2 * X1(KK)
            W3(KK) = W3(KK) + MU * E3 * X1(KK)
            W4(KK) = W4(KK) + MU * E4 * X1(KK)
         ENDDO
      ENDIF

      IF(IVAL .EQ. 2) THEN
         DO KK = 1,NEL
            W1(KK) = W1(KK) + MU * E1 * X2(KK)
            W2(KK) = W2(KK) + MU * E2 * X2(KK)
            W3(KK) = W3(KK) + MU * E3 * X2(KK)
            W4(KK) = W4(KK) + MU * E4 * X2(KK)
         ENDDO
      ENDIF

      IF(IVAL .EQ. 3) THEN
         DO KK = 1,NEL
            W1(KK) = W1(KK) + MU * E1 * X3(KK)
            W2(KK) = W2(KK) + MU * E2 * X3(KK)
            W3(KK) = W3(KK) + MU * E3 * X3(KK)
            W4(KK) = W4(KK) + MU * E4 * X3(KK)
         ENDDO
      ENDIF

      IF(IVAL .EQ. 4) THEN
         DO KK = 1,NEL
```

```
          W1(KK) = W1(KK) + MU * E1 * X4(KK)
          W2(KK) = W2(KK) + MU * E2 * X4(KK)
          W3(KK) = W3(KK) + MU * E3 * X4(KK)
          W4(KK) = W4(KK) + MU * E4 * X4(KK)
       ENDDO
    ENDIF

C---   CHECK TO SEE IF YOU'RE DONE

       IF (ITER .EQ. ITERLIM) GOTO 999

       GOTO 10

C---   YOU'RE DONE. WRITE OUT FINAL WEIGHTS

 999   WRITE(6,100)ITER
       TYPE *,' '


       DO KK = 1,NEL
          WRITE(6,191) W1(KK), W2(KK), W3(KK),W4(KK)
       END DO

 191   FORMAT( 4(2X,G,3X))

       TYPE *,' '

 100   FORMAT(2X,'ITERATION: ',I7)
 102   FORMAT ( 5(2X,F14.7,5X))

C---   AGAIN?

       WRITE(6,133)
 133   FORMAT(2X,'AGAIN? [Y/N]: ',$)
       READ(6,144)AGNFLG
 144   FORMAT(A1)

       IF (AGNFLG .EQ. 'N') GOTO 1001

       GOTO 73

 1001  OPEN(UNIT = 23,
       9     NAME = 'WEIGHTS.LMS',
       9     STATUS = 'NEW',
       9     ACCESS = 'SEQUENTIAL')

       DO KK = 1,NEL
          WRITE(23,191)W1(KK),W2(KK), W3(KK), W4(KK)
       END DO

       WRITE(23,761)ITERLIM,MU
 761   FORMAT(I,10X,G)
       CLOSE(23)


       STOP
       END
```

```
C---    THIS PROGRAM USES THE OUTPUT OF THE LMS ALGORITHM TO
C---    CLASSIFY FEATURE VECTORS

        PROGRAM    THECLASS

        REAL                  X1(9)
        REAL                  W1(9),     W2(9),      W3(9),W4(9),W5(9)
        REAL                  SUM(5)
        REAL                  MAXVAL
        BYTE                  AGN
        CHARACTER*32          FNAME

C---    GET WEIGHT VECTORS FROM FILES

 20     WRITE(6,734)
 734    FORMAT(2X,'ENTER NUMBER OF ELEMENTS IN FEATURE VECTORS: ',$)
        READ(6,735)NEL
 735    FORMAT(I)

        WRITE(6,750)
 750    FORMAT(2X,'ENTER NAME OF FILE OF WEIGHTS: ',$)
        READ(6,751) FNAME
 751    FORMAT(A)


        OPEN(UNIT = 3,
        9    NAME = FNAME,
        9    STATUS = 'OLD',
        9    ACCESS = 'SEQUENTIAL')

        DO KK = 1,NEL
           READ(3,1)W1(KK),W2(KK),W3(KK),W4(KK)
        END DO

        CLOSE (3)

 1      FORMAT(  4(2X,G,3X))

        DO KK = 1,NEL
           WRITE(6,1)W1(KK), W2(KK), W3(KK), W4(KK)
        END DO

C---    INPUT UNKNOWN DATA VECTOR

 5595   WRITE(6,881)
 881    FORMAT(2X,'ENTER FILENAME OF UNKNOWN FEATURE VECTOR: ',$)
        READ(6,882) FNAME
 882    FORMAT(A)

        OPEN(UNIT= 3,
        9    NAME = FNAME,
        9    STATUS = 'OLD',
        9    ACCESS = 'SEQUENTIAL')

        DO KK = 1,NEL
           READ(3,2)X1(KK)
        END DO

 2      FORMAT(G)

        DO KK = 1,NEL
           WRITE(6,2)X1(KK)
        END DO

        DO KK = 1,4
           SUM(KK) = 0.0
        ENDDO

        DO KK = 1,NEL
           SUM(1) = W1(KK) * X1(KK) + SUM(1)
           SUM(2) = W2(KK) * X1(KK) + SUM(2)
           SUM(3) = W3(KK) * X1(KK) + SUM(3)
           SUM(4) = W4(KK) * X1(KK) + SUM(4)
```

```
      END DO

C---  FIND LARGEST ELEMENT IN SUM VECTOR

      MAXVAL = -100000.

      DO KK = 1,4
         IF(SUM(KK) .GT. MAXVAL) THEN
            MAXVAL = SUM(KK)
            IND = KK
         ENDIF
      ENDDO

C---  WRITE RESULTS

      WRITE(6,5)IND,SUM(IND)
 5    FORMAT(2X,'UNKNOWN BELONGS TO CLASS ',I1,10X, 'SUM IS: ',G)

      TYPE *,' '
      TYPE *,' '

      WRITE(6,1999)SUM(1),SUM(2),SUM(3),SUM(4)
 1999 FORMAT( 4(2X,G,3X))

C---  DO YOU WANT TO INPUT ANOTHER UNKNOWN VECTOR?

      WRITE (6,50)
 50   FORMAT(2X,'AGAIN? [Y/N]: ',$)
      READ(6,51)AGN
 51   FORMAT(A1)

      IF(AGN .EQ. 'N') THEN
         GOTO 99
      ELSE
         GOTO 5595
      ENDIF

 99   WRITE(6,1998)
 1998 FORMAT(2X,'SUM1',15X,'SUM2',15X,'SUM3',15X,'SUM4')

      TYPE *,' '


      STOP
      END
```

VITA

Mr. Martin P. DeSimio was born on 3 April 1960 in Burbank, California. He graduated from high school in Fairborn, Ohio in 1978 and then attended Wright State University in Dayton, Ohio. He graduated with the degree of Systems Engineer, Electrical Option in June 1983. Then, he accepted a position with the Foreign Technology Division at Wright-Patterson Air Force Base, Ohio as an Electronics Engineer in the Directorate of Sensor Data. He entered the School of Engineering, Air Force Institute of Technology, in June 1986.

Permanent address: 4770 Appaloosa Trail
Fairborn, OH 45324-9700

END
DATE
FILMED
8-88
DTIC